



Universidade de Aveiro
2011

Departamento de Electrónica, Telecomunicações e
Informática

**Aílton Marques
Lopes Tolentino**

**Interface para a Integração entre o OpenERP e o ERP
Primavera**



Universidade de Aveiro
2011

Departamento de Electrónica, Telecomunicações
e Informática

**Aílton Marques
Lopes Tolentino**

Interface para a Integração entre o OpenERP e o ERP Primavera

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Electrónica e Telecomunicações – Especialização em Sistemas de Informação, realizada sob a orientação científica do Doutor Joaquim Sousa Pinto, Professor Auxiliar do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro

Com o apoio da Cooperação Portuguesa



**COOPERAÇÃO
PORTUGUESA**

o júri

Presidente

Prof. Doutor José Manuel Matos Moreira
Professor Auxiliar da Universidade de Aveiro

Vogais:

Prof. Doutor Joaquim Manuel Henriques de Sousa Pinto
Professor Auxiliar da Universidade de Aveiro

Prof. Doutor Carlos Alberto Baptista de Sousa Pinto
Professor Auxiliar do Departamento de Sistemas de Informação da Escola de Engenharia da
Universidade do Minho

agradecimentos

Gostaria de agradecer a todos os que, directa ou indirectamente, me dedicaram algum do seu tempo, em especial:

Ao meu orientador, professor Joaquim Sousa Pinto, pelo seu apoio, disponibilidade e confiança demonstrados durante a realização deste trabalho.

À minha esposa, Ermelinda Cardoso, pela sua paciência, companheirismo e motivação transmitida.

À minha família, pelo carinho e apoio.

Aos meus colegas de turma, pela sua cumplicidade, amizade e partilha de conhecimentos ao longo da realização do Mestrado.

palavras-chave

Tecnologias de Informação, Sistemas de Informação, ERP, Primavera BSS, OpenERP, *Software* Livre, Integração de Sistemas.

resumo

O fenómeno da globalização fez com que muitas empresas passassem a operar em vários mercados. Com a crescente competitividade dos mercados, a gestão da informação transformou-se num factor crucial para alcançar o sucesso.

Perante esta realidade, várias são as empresas que têm adoptado *softwares* integrados de gestão (*Enterprise Resource Planning* - ERP), com o objectivo de otimizar o tratamento da informação, bem como de conceber ferramentas eficazes de análise de informação para tomada de decisão.

Paralelamente, a integração de sistemas de informação passou a ser um factor diferenciador, permitindo responder rápida e eficazmente às necessidades do mercado.

Em Cabo Verde, o ERP Primavera tem uma elevada taxa de utilização mas, ultimamente têm surgido novas alternativas, algumas delas baseadas em soluções *Open Source*, onde se destaca o OpenERP. Algumas empresas já utilizam essas duas soluções em simultâneo, procurando conciliar as principais funcionalidades de cada uma delas.

É neste âmbito que o presente projecto se insere, tendo como principal objectivo o desenvolvimento de uma solução que seja capaz de integrar o OpenERP e o ERP Primavera, de modo a aproveitar os pontos fortes de cada um deles, possibilitando assim a integração e automatização dos processos de negócio.

keywords

Information technology, Information Systems, ERP, Primavera BSS, OpenERP, Open Source, Systems Integration.

abstract

The phenomenon of globalization has caused many companies to engage operation in various markets. The ever growing competitiveness in markets has transformed information management into a critical success factor.

Taking this reality into consideration, many companies have adopted Integrated Management Software/Enterprise Resource Planning (ERP) with the objective of maximizing information processing as well as conceiving efficient information analysis tools for decision-making.

Concurrently, the integration of information systems has become a differentiating factor which permits for rapid and efficient responses to market requirements.

In Cape Verde, ERP Primavera is highly utilized, however, new alternatives have recently surged, some of which are based on Open Source solutions, among which OpenERP stands out. Some companies already use these two solutions simultaneously with the aim of reconciling the main functionalities of each one.

This project is integrated under this scope, having as its main objective the development of a solution that is able to integrate OpenERP and ERP Primavera so as to take advantage of each of their strong points and thus allowing for the integration and automation of business procedures.

Índice

ÍNDICE.....	I
ÍNDICE DE FIGURAS.....	III
ÍNDICE DE TABELAS.....	V
ABREVIATURAS	VI
1 INTRODUÇÃO	1
1.1 Objectivos propostos	2
1.2 Organização da dissertação	3
2 FUNDAMENTOS TEÓRICOS.....	4
2.1 Sistemas de Informação.....	4
2.1.1 Conceitos e Funções de um Sistema de Informação.....	4
2.1.2 Dimensões de um Sistema de Informação.....	7
2.1.3 Principais categorias de Sistemas de Informação	9
2.2 Enterprise Resource Planning (ERP)	13
2.2.1 Características fundamentais.....	15
2.2.2 Componentes típicos.....	15
2.2.3 Vantagens e desvantagens.....	16
2.2.4 Implementação	17
2.3 Integração de Sistemas de Informação	19
2.3.1 Motivações e evolução da integração de SI	19
2.3.2 Conceito de Enterprise Application Integration (EAI)	22
2.3.3 Âmbitos de integração	23
2.3.4 Tipos de integração	24
2.3.5 Tecnologias de integração.....	28
2.4 Caso de Estudo: Integração entre SAP e Amadeus	38
2.4.1 Características e arquitectura do Sistema Integrado SAP R/3	39
2.4.2 SAP Netweaver	41
2.4.3 Âmbito da integração entre o SAP Travel Management e o Amadeus e-Travel Management	42
3 FERRAMENTAS.....	46
3.1 Primavera Business Software Solutions	46
3.1.1 Características e Arquitectura do ERP Primavera	46
3.1.2 Visual Basic for Applications (VBA)	51
3.2 OpenERP.....	52
3.2.1 Caracterização da solução.....	52
3.2.2 Arquitectura do OpenERP	55

4	ARQUITECTURA PARA A INTEGRAÇÃO OPENERP – PRIMAVERA.....	57
4.1	Arquitectura da solução.....	57
4.2	Descrição do Processo de Integração	60
4.3	Especificação da estrutura dos ficheiros XML.....	62
4.3.1	Ficheiro de Artigos	62
4.3.2	Ficheiro de Clientes	63
4.3.3	Ficheiro de Facturação	64
4.3.4	Ficheiro de Liquidações	66
4.4	Ferramentas Utilizadas	67
4.4.1	ERP Primavera: Campos e Tabelas do Utilizador	68
4.4.2	ERP Primavera: Funções do Utilizador	69
4.4.3	ERP Primavera: Menus do Utilizador	69
4.5	Estrutura das Pastas da Interface.....	70
4.6	Estrutura da Tabela de Parametrização	70
4.7	Resultados obtidos	71
4.7.1	Operações a serem realizadas no OpenERP.....	71
4.7.2	Operações a serem realizadas no ERP Primavera	75
5	CONCLUSÕES E TRABALHO FUTURO	80
5.1	Considerações.....	80
5.2	Trabalho Futuro	81
	REFERÊNCIAS BIBLIOGRÁFICAS	82

Índice de Figuras

Figura 1: Funções de um SI (Adaptado de Laudon e Laudon, 2006)	4
Figura 2: Relação “Dados/Informação/Conhecimento” (Adaptado de Teixeira, 2008)	6
Figura 3: Dimensões de um Sistema de Informação (Adaptado de Laudon e Laudon, 2006)	7
Figura 4: Níveis organizacionais numa empresa (Adaptado de Seruca e Teixeira, 2006)	10
Figura 5: Estrutura típica de um sistema ERP	14
Figura 6: Integração de Sistemas de Informação (Adaptado de Silva, 2003)	21
Figura 7: Evolução da integração de SI (Adaptado de Martins, 2006)	21
Figura 8: Esquema de integração EAI (Fonte: http://www.fujitsu.com)	22
Figura 9: Exemplo de um ficheiro XML	29
Figura 10: Monitor Transaccional (Adaptado de Martins, 2006)	30
Figura 11: Remote Procedure Call	31
Figura 12: Descrição geral da arquitectura MOM	32
Figura 13: Arquitectura de um <i>Message Broker</i>	33
Figura 14: <i>Object Request Broker</i> (Adaptado de Martins, 2006)	33
Figura 15: Funcionamento de um <i>Web Service</i> (Fonte: http://www.sinfic.pt)	36
Figura 16: <i>Web Services Stack</i> (Adaptado de Carvalho, 2003)	37
Figura 17: Arquitectura de 3 camadas (Adaptado de Martins, 2006)	37
Figura 18: Arquitectura SOA (Adaptado de Martins, 2006)	38
Figura 19: Módulos do SAP R/3	40
Figura 20: Arquitectura do sistema integrado de gestão de viagens (Fonte: SAP Travel Management (FI-TV)) ..	44
Figura 21: Esquema da Solução Global Primavera (Fonte: Primavera BSS)	46
Figura 22: ERP Primavera - Ambiente de trabalho	47
Figura 23: ERP Primavera - Arquitectura de 3 Níveis	48
Figura 24: ERP Primavera - Arquitectura do Motor	49
Figura 25: ERP Primavera - Propriedades do objecto GcpBEDocumentoVenda	50
Figura 26: Ambiente de Trabalho do OpenERP	54
Figura 27: Arquitectura do OpenERP (Fonte: http://www.openerp.com/services/books)	56
Figura 28: Arquitectura da solução	57
Figura 29: Diagrama Relacional das Vendas no ERP Primavera	58
Figura 30: Diagrama Relacional dos Recebimentos no ERP Primavera	59
Figura 31: Tabelas movimentadas na integração com o módulo de Contabilidade do ERP Primavera	60
Figura 32: Ficheiros gerados pelo OpenERP	61
Figura 33: Importação dos ficheiros no Primavera e integração na Contabilidade	61
Figura 34: <i>XML Schema</i> da entidade Artigo	63
Figura 35: <i>XML Schema</i> da entidade Cliente	64
Figura 36: <i>XML Schema</i> dos documentos de venda	65
Figura 37: <i>XML Schema</i> dos documentos de liquidação	67
Figura 38: Assistente de criação de Tabelas e Campos do Utilizador	68
Figura 39: Assistente de criação de uma Função do Utilizador	69
Figura 40: Criação de Menus de Utilizador	69
Figura 41: Estrutura das pastas	70
Figura 42: Tabela de Utilizador TDU_InterfaceOpenERP	71
Figura 43: Artigos criados no OpenERP	71
Figura 44: Confirmação da criação do cliente C1 (António Silva)	72
Figura 45: Representação da Factura emitida	72

Figura 46: Criação e mapeamento dos Modos de Pagamento	73
Figura 47: Emissão do Recibo do cliente C1.....	73
Figura 48: Formulário de exportação das vendas emitidas	73
Figura 49: Exportação do ficheiro de vendas	74
Figura 50: Conteúdo do ficheiro de vendas exportado pelo OpenERP	74
Figura 51: <i>Layout</i> do integrador desenvolvido	75
Figura 52: Log – Erro na integração de um ficheiro de Artigos.....	76
Fig. 53: Log - Erro na integração de um ficheiro de clientes	76
Figura 54: Ficha de clientes - Preenchimento automático do campo “Conta Contabilidade”	77
Figura 55: Parametrização da criação de contas de Clientes na Contabilidade.....	77
Figura 56: Detalhes da Factura integrada no ERP Primavera.....	78
Figura 57: Apresentação dos dados integrados no ERP Primavera	78
Figura 58: Integração da Factura na Contabilidade	79

Índice de Tabelas

Tabela 1: Campos da entidade Artigo	62
Tabela 2: Campos da entidade Cliente.....	63
Tabela 3: Campos do ficheiro de Facturação	65
Tabela 4: Campos do ficheiro de liquidações.....	66

Abreviaturas

API	Application Programming Interface
ASAP	Accelerated SAP
BSS	Business Software Solutions
CIM	Computer Integrated Manufacturing
COM	Component Object Model
CORBA	Common Object Request Broker Architecture
CRM	Customer Relationship Management
DCOM	Distributed Component Object Model
DLL	Dynamic Link Library
DNA	Distributed InterNet Applications Architecture
EAI	Enterprise Application Integration
ERP	Enterprise Resource Planning
GDS	Global Distribution System
GPL	General Public License
GUI	Graphical User Interface
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
IDE	Integrated Development Environment
IDL	Interface Definition Language
IIOP	Internet Inter-Orb Protocol
J2EE	Java 2 Platform Enterprise Edition
MRP	Materials Requirements Planning
MRPII	Manufacturing Resource Planning
MT	Monitores Transacionais
OLE	Object Linking and Embedding
OMG	Object Management Group
ORB	Object Request Broker
PME	Pequenas e Médias Empresas
PNC	Plano Nacional de Contabilidade
RFC	Remote Function Call
RMI	Remote Method Invocation
RPC	Remote Procedure Call
SCM	Supply Chain Management
SGBD	Sistema de Gestão de Bases de Dados
SI	Sistemas de Informação
SNCRF	Sistema de Normalização Contabilística e do Relato Financeiro
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SOI	Service Oriented Integration
SQL	Structured Query Language
TI	Tecnologias de Informação
TIC	Tecnologias de Informação e Comunicação
UDDI	Universal Description, Discovery and Integration
VBA	Visual Basic for Applications
WS	Web Services

WSDL Web Services Description Language
XML Extended Markup Language

1 Introdução

A gestão da informação é, actualmente, um verdadeiro factor competitivo de sucesso. Com esta realidade, várias são as empresas que adoptam *softwares* integrados de gestão, com o objectivo de facilitar e regular a entrada de informação no sistema, bem como de conceber ferramentas eficazes de análise de informação para tomada de decisão.

As Tecnologias da Informação (TI) correspondem hoje em dia a um dos principais pilares das organizações. A procura do controlo da informação e da flexibilização, tornou a integração de sistemas informáticos uma das grandes prioridades organizacionais. Por seu turno, a constante evolução das TI criou realidades tecnológicas díspares fomentando a necessidade de partilhar informação e funcionalidades entre sistemas. Para as organizações, a questão tecnológica da integração de Sistemas de Informação (SI) tornou-se cada vez mais complexa e é hoje um autêntico desafio quanto à sua flexibilidade, adaptabilidade, implementação, manutenção e gestão. (Martins, 2006)

Com a globalização dos negócios, as empresas passaram a operar em vários mercados. Um dos efeitos da globalização traduz-se, nomeadamente, na possibilidade de as empresas cotadas poderem financiar-se em diferentes mercados bolsistas. Uma empresa que pretenda ter valores cotados numa determinada bolsa deveria apresentar as respectivas contas de acordo com os princípios contabilísticos aceites nesse país. Esta exigência implica um consequente acréscimo de custos administrativos associados à preparação e apresentação das demonstrações financeiras. É neste contexto que surgem os argumentos a favor de uma maior harmonização (normalização) contabilística a nível internacional. (Gonçalves e Fernandes, 2009)

Com a implementação, em Janeiro de 2009, do Sistema de Normalização Contabilística e de Relato Financeiro (SNCRF) em Cabo Verde, várias empresas sentiram a necessidade de recorrer à informatização, adoptando ou actualizando os *softwares* de gestão, de modo a responder aos desafios impostos pelo novo sistema.

De acordo com o Decreto-Lei n.º 5/2008, de 4 de Fevereiro, publicado no Boletim Oficial, I Série, da República de Cabo Verde, o SNCRF, que veio substituir o Plano Nacional de Contabilidade (PNC), tem por objectivo acompanhar os desenvolvimentos havidos nas directivas internacionais quanto à qualidade da informação financeira. Assim, as alterações introduzidas tiveram em atenção a previsível evolução do sistema fiscal, a modernização do tecido económico do País e do mercado de capitais, bem como a preocupação de melhorar a qualidade das

demonstrações financeiras e do relato financeiro das empresas, como factor de credibilidade ao funcionamento da economia real em Cabo Verde.

Grande parte das empresas cabo-verdianas utiliza o *Enterprise Resource Planning* (ERP) da Primavera BSS (*Business Software Solutions*). Outras têm vindo a apostar em soluções *Open Source* (*Software Livre*), área onde o OpenERP tem vindo a ganhar alguma preponderância. Algumas dessas empresas utilizam alguns módulos do OpenERP, mais concretamente o módulo da Gestão Comercial, em áreas como o Turismo, Educação, Imobiliária, Sector Automóvel, etc. No entanto, em relação ao módulo de Contabilidade, a preferência tem recaído sobre a solução da Primavera, por se tratar de um produto que apresenta importantes funcionalidades e que se encontra bem adaptado ao SNCRF. De salientar que o ERP Primavera é o mais utilizado pelos contabilistas cabo-verdianos.

As empresas que utilizam esses dois sistemas, têm a desvantagem de ter de processar os dados relacionados com as vendas, cobranças, encomendas e gestão bancária no OpenERP e, posteriormente, lançar os documentos contabilísticos, de forma manual, no módulo de Contabilidade do ERP Primavera.

1.1 Objectivos propostos

Este trabalho tem como objectivo o desenvolvimento de uma arquitectura que seja capaz de integrar o OpenERP e o ERP Primavera, de modo a aproveitar os pontos fortes de cada um deles. Mais concretamente, propõe-se integrar o módulo Comercial do OpenERP (Gestão de Artigos e Clientes, Vendas e Facturação, Recebimentos e Gestão de Contas Correntes de Clientes) com o módulo de Contabilidade da Primavera.

Com esta solução, pretende-se obter as seguintes vantagens:

- ✓ A integração da informação passa a ser feita de forma automática e, não havendo intervenção humana nesse processo, a probabilidade de ocorrência de erros é menor;
- ✓ Redução dos custos operacionais;
- ✓ Obtenção de uma maior rentabilidade do negócio.

A metodologia utilizada será baseada no estudo do ambiente de desenvolvimento e das principais funcionalidades dos dois sistemas e no desenvolvimento de uma solução que seja capaz de interligar essas duas plataformas distintas.

1.2 Organização da dissertação

No capítulo 1, é feita uma breve introdução, onde são apontados os aspectos que serviram como motivação para a realização deste trabalho, assim como os objectivos inicialmente propostos.

No capítulo 2, são apresentados os fundamentos teóricos do trabalho desenvolvido, começando com uma abordagem aos SI, seguido de uma introdução ao conceito de ERP, assim como a apresentação das suas características, os prós e contras e os cuidados a ter na sua implementação. É também introduzida a temática da Integração de SI, com enfoque nas razões para a integração, bem como nos tipos e tecnologias de integração existentes. Ainda neste capítulo, é apresentado um caso de estudo, envolvendo a integração do ERP da SAP, líder mundial do ramo, com o *software* de gestão de viagens da Amadeus.

O capítulo 3 descreve as ferramentas utilizadas para a integração (ERP Primavera e OpenERP), as suas funcionalidades e motores associados.

O capítulo 4 apresenta a arquitectura de integração entre o OpenERP e o Primavera, as funcionalidades do sistema concebido, as metodologias e tecnologias adoptadas e os resultados obtidos.

O capítulo 5 está reservado à apresentação das conclusões da dissertação, bem como algumas linhas de orientação para o trabalho futuro a desenvolver neste âmbito.

2 Fundamentos Teóricos

Este capítulo tem como principal objectivo a contextualização dos conceitos, termos e abordagens apresentados no âmbito desta dissertação.

2.1 Sistemas de Informação

2.1.1 Conceitos e Funções de um Sistema de Informação

Segundo Laudon e Laudon (2006), um SI pode ser definido tecnicamente, como um conjunto de componentes inter-relacionados que recolhem, processam, armazenam e distribuem informações de apoio à decisão e de controlo numa organização. Além do suporte à tomada de decisão, coordenação, e controlo, os SI auxiliam os gestores e funcionários a analisar problemas, a visualizar soluções e a tomar decisões de vários níveis.

Cassarro (2004) define um SI como uma combinação estruturada de informação, recursos humanos, tecnologias de informação e práticas de trabalho, organizado de forma a permitir o melhor atendimento dos objectivos da organização.

De acordo com Stair e Reynolds (2008), um SI é um conjunto de elementos ou componentes que interagem entre si e participam na recolha de dados (inputs), processamento e disseminação de informação (output), estando sujeito a constantes ajustamentos perante mecanismos de retro-alimentação (feedback).

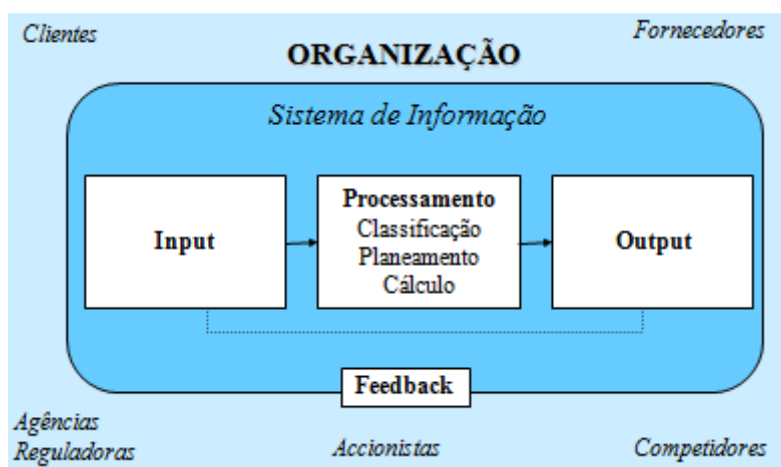


Figura 1: Funções de um SI (Adaptado de Laudon e Laudon, 2006)

Na Figura 1 estão representadas as actividades de um SI (Input, Processamento, Output e Feedback), que foram assim definidas por Stair (2002):

- ✓ **Input** – é a actividade de captar e juntar dados primários. Por exemplo, ao se produzir cheques de pagamento por um sistema informatizado, as horas trabalhadas de cada empregado devem ser informadas antes que o cheque seja efectivamente calculado e emitido. Independentemente do sistema envolvido, o tipo de input é determinado pela saída desejada do sistema;
- ✓ **Processamento** – envolve a conversão ou transformação dos dados em saídas úteis e desejadas pelo utilizador. Um exemplo clássico é o processamento de uma folha de pagamento de salários, onde o salário líquido, as horas extras e os descontos são calculados mediante a multiplicação das horas de trabalho realizadas por cada empregado e do valor do salário hora.
- ✓ **Output** – envolve a etapa na qual a informação propriamente dita é emitida, ou seja, é a etapa que realmente interessa ao utilizador do sistema. Se uma saída gerada por um sistema não for útil para algum propósito, então deve-se fazer uma crítica ao sistema para avaliar sua real necessidade.
- ✓ **Feedback** – é uma saída usada para fazer ajustes ou modificações nas actividades de entrada ou no processamento. Por exemplo, erros de digitação podem fazer com os que dados de entrada tenham que ser corrigidos antes do seu processamento.

Stair e Reynolds (2008) possuem uma visão distinta. Defendem que o mecanismo de feedback é o componente que ajuda as organizações a alcançar as suas metas, como o lucro e a melhoria do serviço ao cliente. As empresas podem utilizar os SI para aumentar as suas receitas e reduzir os seus custos.

Ainda segundo os mesmos autores, para um melhor entendimento do significado e importância de um SI numa organização, é necessário ter em atenção outros conceitos elementares, como por exemplo, dados, informação e conhecimento. Neste contexto, dados são factos em bruto que isolados não têm qualquer significado, como por exemplo o número de empregados, número de horas que trabalham numa semana, número de inventários ou ordens de venda, ao passo que informação é uma colecção dos factos organizados de uma forma que estes tenham valor adicional para além do valor dos próprios factos. Para que a informação seja útil, nomeadamente

para a tomada de decisão, é necessário o conhecimento que não é mais do que a capacidade de entender e inter-relacionar a informação de que dispomos.

Para McGee e Prusak (1994), informação vem da palavra latina *informare*, que significa dar forma a algo. Assim, informações são dados recolhidos, organizados e ordenados, aos quais são atribuídos significados e contexto. A informação é, portanto, o dado trabalhado que permite a tomada de decisões.

Segundo Davenport e Prusak (2000), o conhecimento é um conjunto de informações que incluem reflexão, síntese e contexto. Pode ser visto como um refinamento de informações, a que está associada uma certa dose de inteligência que é capaz de fazer associações entre informações, experiências e conceitos e elaborar conclusões.

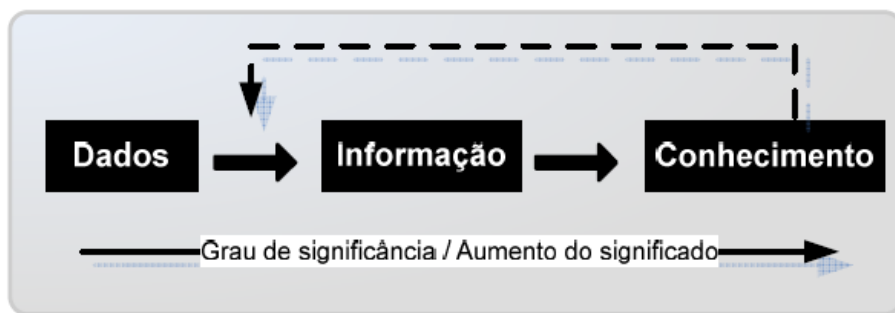


Figura 2: Relação “Dados/Informação/Conhecimento” (Adaptado de Teixeira, 2008)

Mülbert e Ayres (2005) defendem que, se as organizações despendem preciosos esforços e recursos para a implementação de um SI, certamente elas esperam obter algum retorno. Os sistemas bem-sucedidos devem proporcionar ganhos que, no mínimo, devem equivaler aos esforços despendidos.

De seguida é apresentado um conjunto de características que, embora possam ser difíceis de medir, podem ser utilizadas objectiva ou intuitivamente para aferir o sucesso da implementação de um SI:

- ✓ **Rentabilidade** – existe quando os benefícios do sistema ultrapassam os custos associados;
- ✓ **Desempenho** – ocorre quando o sistema melhora a qualidade das decisões dos utilizadores;

- ✓ **Áreas de aplicação** – um sistema é bem sucedido quando é aplicado aos problemas de maior importância na organização, o que contribui significativamente para justificar o seu custo;
- ✓ **Satisfação dos utilizadores** – como o sistema é um instrumento de auxílio ao utilizador, a satisfação deste indica que o sistema corresponde às expectativas criadas;
- ✓ **Utilização generalizada** – o facto de o sistema ser amplamente utilizado é um indicador de sucesso na medida em que possui a aprovação de várias pessoas.

2.1.2 Dimensões de um Sistema de Informação

De acordo com Laudon e Laudon (2006), o uso eficiente dos SI requer uma compreensão da organização, gestão e tecnologias (*hardware, software, bases de dados, comunicações, etc.*). Um SI é um produto destas três dimensões. (Figura 3)



Figura 3: Dimensões de um Sistema de Informação (Adaptado de Laudon e Laudon, 2006)

Um SI cria valor para a empresa como uma solução de gestão e organização quando desafiadas pelo ambiente externo.

Cada dimensão tem a sua particularidade no contexto dos SI, como se detalha a seguir:

Organização

Gonçalves (2000) defende que as organizações podem ser vistas como uma grande colecção de processos operacionais e administrativos, onde os processos operacionais criam, produzem e

entregam os bens e serviços que são consumidos pelo mercado, enquanto os processos administrativos são responsáveis pelo planeamento e controle da condução dos negócios.

Esses processos seguem uma série de procedimentos que podem estar formalizados e escritos, ou compor práticas de trabalho informais.

Gestão

Segundo Laudon e Laudon (2006), citados por Amaral (2009, p. 11), o trabalho de gestão está associado ao processo de tomada de decisão e formulação de planos de acção para resolver problemas organizacionais. Os gestores tentam perceber os desafios colocados pelo ambiente empresarial, determinam a estratégia organizacional para responder a esses desafios e atribuem os recursos humanos e financeiros para coordenar os trabalhos e alcançar o sucesso, exercendo uma liderança responsável. No entanto, os gestores devem fazer mais do que gerir o que já existe, eles devem também criar novos produtos e serviços e até reorganizar a organização de tempos em tempos. Uma parte substancial da responsabilidade de gestão é o trabalho criativo impulsionado por novos conhecimentos e nova informação. As Tecnologias da Informação e de Comunicação (TIC) podem desempenhar um papel importante na reorganização e redesenho da organização.

Tecnologia

A tecnologia é o meio pelo qual os SI podem ser implementados. As TIC são ferramentas bastante úteis para lidar com a mudança, constituindo a base de suporte à criação de SI nas organizações. São constituídas pelo computador propriamente dito e demais equipamentos (*hardware*), pelos programas de computadores (*software*), pelas tecnologias de armazenamento para organizar e armazenar os dados (bases de dados) e pelos recursos de telecomunicações que interconectam os computadores em rede, possibilitando a partilha de dados.

Para Laudon (1994), citado por Ribeiro (2009, p. 17), toda a infra-estrutura de informação (desde computadores, *softwares*, máquinas, etc.), que cada organização criar, trará poder e acesso a recursos extremamente importantes, sendo a rapidez no acesso à informação um dos factores determinantes, e até vital, para a sobrevivência das organizações no mercado. Segundo uma perspectiva mais organizacional, é como uma combinação de procedimentos, informação, pessoas e tecnologias de informação, organizadas para o alcance dos objectivos de uma organização. A vantagem desta tecnologia está na capacidade de integração da informação, tanto a nível interno da organização como na integração da informação de múltiplas organizações.

2.1.3 Principais categorias de Sistemas de Informação

Para Mülberty e Ayres (2005) não existe um único grande sistema que atenda a todas as necessidades de informação existentes numa empresa. O que se verifica é a existência de diferentes tipos de sistemas de informação para atender os diferentes níveis de problemas, as diferentes funções existentes dentro delas e até mesmo as diferentes estruturas organizacionais.

Segundo Rezende e Abreu (2001), citados por Mülberty e Ayres (2005, p. 54), uma primeira forma de classificar os SI é pela sua amplitude na estrutura organizacional. Os SI podem ser construídos para sedes, divisões, departamentos, equipas específicas e até pessoas físicas. Podem ainda abranger outras organizações (parceiros de negócios), ou mesmo instalações localizadas em outros países (multinacionais).

Para Laudon e Laudon (2006), uma vez que existem diferentes tipos de interesses, áreas de actuação e níveis estruturais numa organização, existem também diferentes tipos de SI, podendo ser classificados segundo vários critérios. Nenhum sistema isolado consegue, hoje em dia, providenciar toda a informação que uma organização necessita, tendo por isso o conceito evoluído juntamente com o termo integração.

Ainda segundo os mesmos autores, citados por Seruca e Teixeira (2006), os critérios poderão ser divididos nos seguintes níveis:

- ✓ **Nível Operacional (Transaccional)** – suporta directamente operações quotidianas de uma organização, ajudando os gestores no acompanhamento de actividades transaccionais e elementares tais como vendas, recibos, depósitos de dinheiro, folhas de pagamento, decisões de crédito, etc. O principal objectivo dos sistemas a este nível é responder a questões de rotina, como por exemplo: “Quantas peças existem em stock?”, “Qual foi o modo de pagamento utilizado por determinado cliente?”.
- ✓ **Nível Tático** – suporta decisões com impacto no curto prazo, como por exemplo, análise da qualidade. São vários os autores que defendem a existência de um nível tático que integra os níveis de gestão e do conhecimento:
 - **Nível do Conhecimento** – oferece suporte ao negócio para integrar novas tecnologias e auxiliar a organização no controlo dos fluxos de documentos.
 - **Nível de Gestão** – serve para as actividades de monitorização, tomada de decisão, controlo e de gestão a nível intermédio. Normalmente fornece relatórios periódicos, em vez de informação instantânea sobre as operações. Os sistemas a

esse nível, suportam decisões não rotineiras com tendência a ajudarem em decisões menos estruturadas. Respondem frequentemente a perguntas do tipo: “E se?”, “Qual seria o impacto na produção se dobrássemos as vendas no mês de Dezembro?”, “O que aconteceria à rentabilidade de investimento se a produção atrasasse seis meses?”. As respostas a estas perguntas exigem frequentemente novos dados, muitas vezes do ambiente externo da organização, para além dos dados internos que não podem ser facilmente mostrados pelos sistemas do nível operacional.

- ✓ **Nível Estratégico** – os sistemas do nível estratégico auxiliam a gestão de topo a manipular, abordar e endereçar questões estratégicas e tendências a longo prazo, ambas na organização e no ambiente externo. O seu principal objectivo é conciliar alterações no ambiente externo com a capacidade organizacional existente, encontrando respostas a questões como: “Que produtos devem ser criados em cinco anos?”, “Quais serão os níveis do emprego em cinco anos?”, etc.



Figura 4: Níveis organizacionais numa empresa (Adaptado de Seruca e Teixeira, 2006)

Laudon e Laudon (2006) defendem ainda que, para além das classificações em termos de níveis organizacionais, os SI podem também ser classificados pelas seguintes áreas funcionais, já que servem as funções de negócio principais:

- ✓ **Comercial e Marketing** – estes sistemas acompanham as tendências de vendas, monitorizam o desempenho dos concorrentes, fornecem suporte a pesquisas de mercado, campanhas promocionais e decisões quanto a preços; permitem análises de desempenho das vendas e dos vendedores; ajudam na localização e contacto com potenciais clientes, entre outras funções.

- ✓ **Finanças e Contabilidade** – estabelecem objectivos de investimentos a longo prazo e fornecem previsões do desempenho financeiro da empresa; ajudam a visualizar e controlar os recursos financeiros; monitorizam o fluxo de caixa, contas a receber e a pagar; e emitem relatórios de balanço e mapas fiscais.
- ✓ **Produção** – os sistemas de produção tratam do planeamento, desenvolvimento e manutenção das instalações de produção; do estabelecimento dos objectivos de produção; da aquisição, armazenamento e disponibilidade dos materiais de produção; e do planeamento dos equipamentos, instalações, materiais e mão-de-obra necessários para a embalagem dos produtos acabados.
- ✓ **Recursos Humanos** – tratam da concepção e execução de estratégias de gestão de recursos humanos e tradicionalmente têm as seguintes funções: planeamento (definição de metas e estratégias); organização (determinar o quê/quem/como/para quem/onde); liderança (motivar, dirigir, resolver conflitos) e controlo (monitorizar o desempenho em relação ao planeamento).

De acordo com Laudon e Laudon (2006) e O'Brien e Marakas (2008), citados por Teixeira (2008, p. 22), uma das tipologias de classificações mais utilizadas baseia-se em cinco categorias de SI:

1. **Sistemas de Processamento de Transacções (SPT)**

Tiveram origem nos anos cinquenta, com o aparecimento dos primeiros computadores. Realizam as operações diárias de uma organização, também designadas por transacções. Este tipo de SI é o maior produtor de informações e é requisitado pelos outros sistemas. O principal objectivo destes sistemas é realizar operações rotineiras, permitindo assim a redução do tempo que a empresa perde diariamente com tarefas repetitivas.

2. **Sistemas de Informação para a Gestão (SIG)**

Tendo surgido na década de sessenta, este tipo de SI proporciona aos gerentes relatórios e consultas sobre o desempenho actual e registos históricos da empresa, de forma a apoiar as actividades de planeamento, controlo e tomada de decisões. Em suma, os SIG fornecem resumos sobre as operações básicas (transacções operacionais) da empresa. Os dados de transacções básicas, arquivados pelos SPT, são agrupados (ou sintetizados) e apresentados

num formato pré-estabelecido. Ou seja, apesar de apresentarem algumas características funcionais em comum com os SPT, estão num patamar superior.

3. **Sistemas de Apoio à Decisão (SAD)**

Os sistemas de apoio à decisão tiveram origem na década de 70 e, diferentemente dos SIG, têm por objectivo dar suporte a decisões menos rotineiras e estruturadas, e não facilmente especificadas com antecipação. Os SAD fornecem suporte computacional interactivo durante o processo de tomada de decisão e ainda avaliam as alternativas através de técnicas do tipo *what-if* (i.e., se eu fizer isto, o que acontecerá?).

4. **Sistemas de Informação para Executivos (SIE)**

Os SIE surgiram nos anos 80 e são os sistemas utilizados pelos gestores de nível mais elevado dentro da hierarquia das organizações. Estes sistemas fornecem acesso rápido e de forma bastante amigável, fazendo uso intensivo de recursos gráficos (cores, símbolos, ícones, botões, imagens e gráficos), e recorrendo ao uso de recursos multimédia.

Para além das quatro categorias de SI acima apresentadas – SPT, SIG, SAD e SIE – que foram desenvolvidas a partir da função original de processamento de dados, nos anos 80, com o surgimento dos computadores pessoais e com o rápido desenvolvimento das TIC, surgiram os SAE (Sistemas de Automação de Escritório).

5. **Sistemas de Automação de Escritório (SAE)**

Este tipo de SI situa-se no nível de conhecimento e tem o objectivo de apoiar as actividades de escritório, contribuindo para aumentar as comunicações e a produtividade de equipas e grupos de trabalho.

Seruca e Teixeira (2006) defendem que, não obstante a possibilidade de os SI de diferentes áreas funcionais poderem trocar informação entre si, seria muito mais vantajoso para as empresas se existisse integração entre esses sistemas, o que originaria uma maior fluidez da informação dentro da organização.

Uma solução para esta questão prende-se com a aquisição de Aplicações Empresariais, que são sistemas que permitem coordenar actividades, decisões e conhecimento, através das diversas áreas funcionais e níveis da organização.

Os ERP, enquadrados neste contexto, serão abordados na secção seguinte.

2.2 Enterprise Resource Planning (ERP)

Enterprise Resource Planning (ERP), em português denominado de Sistema Integrado de Gestão Empresarial, é um sistema que possibilita um fluxo de informação contínuo e consistente por toda a empresa, através de uma base de dados integrada. É um instrumento para a melhoria de processos de negócios, como produção, compras ou distribuição, com informações on-line e em tempo real. Dessa forma, procura-se evitar redundâncias e inconsistências de dados, assegurando-se a integridade do fluxo de informações.

Na década de 70 do século XX, desenvolveu-se (Reid & Sanders, 2002) uma importante aplicação informática para o Planeamento de Necessidade de Materiais (MRP – “*Materials Requirements Planning*”). Os conceitos de Produção Integrada por Computador (CIM – “*Computer Integrated Manufacturing*”) e Planeamento de Necessidade de Recursos (MRP II – “*Manufacturing Resource Planning*”) desenvolvidos nas décadas de 80 e 90, espalharam a ideia de integração dos SI da empresa. Porém, uma série de dificuldades de ordem prática e tecnológica, não permitiram que esta visão fosse implementada em grande parte das empresas.

O consecutivo melhoramento das TI e a redução constante no preço do *hardware* permitiu que as pequenas e médias empresas (PME) pudessem ter acesso a sistemas ERP. Os primeiros sistemas ERP foram desenvolvidos apenas para funcionar nos grandes *mainframes*. Com a entrada em cena dos computadores pessoais, o advento das arquitecturas cliente/servidor de múltiplas camadas em *Unix*, *AS/400* e *Windows NT*, a relação com sistemas de gestão de bases de dados relacionais e integração das mesmas com tecnologias *Web*, contribuiu fortemente para a facilidade de utilização dos ERP.

Assumiu-se que as potenciais empresas clientes deste tipo de produto estivessem distribuídas por escritórios, fábricas, lojas, postos de distribuição, em diferentes espaços físicos, bem como múltiplos pontos de controlo e gestão. Logo, a transferência electrónica de dados pelos diversos locais é bastante importante e muitas vezes vital para muitas empresas. Para facilitar todas estas transacções, sistemas como o *Workflow*, *Workgroups*, *Groupware*, *Internet*, *Intranet*, *Datamining* e *Datawarehousing*, influenciaram de forma decisiva os ERP com vista à caminhada e evolução para o futuro.

Um facto importante é o de, ao integrarem todos os subsistemas existentes numa empresa, os ERP conseguirem resultados melhores que o total dos subsistemas em separado. Os sistemas de aplicação tradicionais que geralmente as empresas usam, tratam todas as transacções

isoladamente. Elas são feitas e usadas para dar resposta a funções específicas para as quais foram destinadas. Os ERP deixaram de olhar para uma transacção como sendo um processo isolado, tratando-a como parte de todo um conjunto de processos interligados que perfazem toda a existência de uma empresa.

A Figura 5 apresenta uma possível estrutura para o funcionamento de um sistema ERP, constituída por diversos módulos que suportam várias actividades de uma empresa, ligados a uma base de dados central partilhada.



Figura 5: Estrutura típica de um sistema ERP

A maior parte dos sistemas de aplicações não integrados, nada mais são que ferramentas de manipulação de dados. Armazenam dados, processam-nos e apresentam-nos de forma apropriada sempre que são requisitados pelo utilizador. Neste processo, o único problema é não haver ligação aparente entre os diversos sistemas usados pelos diferentes departamentos. Um sistema ERP faz a mesma coisa, mas de uma maneira diferente. Antes da introdução dos sistemas ERP, surgiam, facilmente, centenas de tabelas cheias de dados gerados como resultado de diversas transacções, ficando apenas acessíveis no departamento que lhes deu origem, em vez de serem tratados e integrados de forma a serem utilizados por múltiplos utilizadores em múltiplos departamentos e para diferentes fins em que não apenas o objectivo inicial.

2.2.1 Características fundamentais

Haberkorn (1999) defende que os sistemas ERP não servem apenas para integrar os vários organismos de uma empresa. Para verdadeiramente ser considerado ERP, é fundamental que o sistema possua algumas das seguintes características:

- ✓ **Flexibilidade** – um sistema ERP é flexível, de forma a responder às constantes transformações das empresas. A tecnologia cliente/servidor permite que o sistema opere sobre diferentes bases de dados.
- ✓ **Modularidade** – é um sistema de arquitectura aberta, isto é, pode usar um módulo livremente sem que este afecte os restantes. O sistema suporta plataformas múltiplas de *hardware* pois muitas empresas possuem sistemas heterogéneos. Deve também facilitar a expansão e/ou adaptabilidade de mais módulos posteriormente.
- ✓ **Abrangência** – o sistema está apto a suportar diferentes estruturas organizacionais das empresas, bem como uma vasta área negócios.
- ✓ **Conectividade** – o sistema não se deve confinar ao espaço físico da empresa mas permitir a ligação com outras entidades pertencentes ao mesmo grupo empresarial.
- ✓ **Representação da realidade** – deve permitir a simulação da realidade da empresa em computador. De forma alguma, o controlo do sistema deve estar fora do processo de negociação e deve ser possível a elaboração de relatórios para os utilizadores que controlam o sistema.

A principal característica de um ERP é a integração da informação, inserida uma única vez nos seus módulos e imediatamente disponível em todo o sistema, organizada de forma a poder ser utilizada em tempo real nos vários centros de decisão da organização, melhorando a qualidade global da informação disponível. (Azevedo e Serdeira, 2001)

2.2.2 Componentes típicos

Como foi referido, um sistema ERP é constituído por vários módulos, cada um representando uma área funcional ou funções específicas da empresa. Estes módulos são parametrizáveis em função de um cliente específico. Alguns fornecedores têm o seguinte conjunto de módulos/aplicações base:

- ✓ Vendas e Distribuição;
- ✓ Tesouraria;
- ✓ Recursos Humanos;
- ✓ Contabilidade;
- ✓ Gestão de Stocks;
- ✓ Gestão de Produção;
- ✓ Gestão de Projectos;
- ✓ Gestão Financeira.

2.2.3 Vantagens e desvantagens

Segundo um estudo realizado pela Deloitte Consulting em 1998, os benefícios de utilização de sistemas ERP, são:

- ✓ Redução do número de documentos em papel, disponibilizando consulta e introdução *on-line* de informação;
- ✓ Acesso à informação tratada, muito mais rapidamente;
- ✓ Informação detalhada vinda de várias áreas da empresa, o que permite aos auditores criar apresentações com mais conteúdo;
- ✓ Melhor controlo de custos;
- ✓ Menor tempo na resposta e assistência ao cliente;
- ✓ Maior eficiência na cobrança de dívidas e maior controlo na satisfação de notas de encomenda ou abatimentos de notas de crédito;
- ✓ Melhor monitorização do sistema e rápida consulta às bases de dados;
- ✓ Possibilidade de uma rápida alteração nas operações de negócio e ajustes às condições do mercado;
- ✓ Existência de uma base de dados única, para ser utilizada por todas as aplicações, eliminando redundâncias e garantindo a fidelidade e consistência dos dados;
- ✓ Acesso à informação e/ou gestão de um processo, num qualquer terminal de qualquer área do sistema.

Ainda de acordo com o mesmo estudo, não há nada que não tenha o seu lado menos positivo, por isso os seguintes pontos negativos são apontados aos ERP:

- ✓ Custos muito elevados;
- ✓ Dependência de um único fornecedor de *software*;

- ✓ Resistência à mudança: pode haver dificuldade na implementação devido à reorganização da empresa;
- ✓ Tempo de implementação (uma implementação pode durar desde 6 meses até 4-5 anos);
- ✓ Duração da verificação de resultados (verificar os benefícios da introdução do sistema, é um processo que demora cerca de oito meses a um ano após a implementação);
- ✓ Risco: no caso de fracasso na implementação, o resultado pode ser desastroso (algumas empresas chegaram mesmo a abrir falência).

2.2.4 Implementação

A implementação de um ERP já foi comparada a um transplante cardíaco: basicamente removem-se os sistemas de informação de um negócio e efectua-se a sua substituição por sistemas completamente novos. A operação e o recobro são lentos e dolorosos e se o transplante for rejeitado, o paciente pode morrer. No entanto, se não houver transplante, o mais provável é que o doente morra a curto ou médio prazo. A implementação é uma decisão estratégica vital, que tem de ser tomada mais cedo ou mais tarde.

Algumas empresas sugerem metodologias ágeis para serem aplicadas na fase da implementação. É o caso da empresa SAP, por exemplo, que tem a metodologia chamada ASAP¹ (*Accelerated SAP*²), cujo objectivo é diminuir o tempo de implementação dos projectos numa organização.

De seguida, passa-se a descrever, resumidamente, as fases que compõem a implementação de um ERP: (Esteves & Jorge, 2001)

A fase inicial é o planeamento do projecto. Nela é definido o âmbito da implementação, isto é, os processos empresariais que serão inseridos no ERP e, conseqüentemente, os módulos que serão implementados; a estratégia de implementação; os recursos necessários: equipas da própria empresa, consultores, salas, o *software*, *hardware*, etc., e, por fim, o plano temporal do projecto e a data da entrada em produção.

Na fase seguinte, deve ocorrer a formação das equipas no *software* escolhido. Esta formação é, na maioria dos casos, uma visão macro de como o sistema funcionará.

¹ ASAP é uma expressão inglesa que significa “*As Soon As Possible*” – o mais rápido possível, em português.

² ASAP (*Accelerated SAP*) é uma solução que possibilita uma implementação eficiente e uma optimização contínua do software SAP.

Quando se inicia a execução dos trabalhos, o primeiro passo é o levantamento dos processos empresariais correntes, seguido da reestruturação destes processos para aumento da sua eficiência, considerando-se as melhorias a serem introduzidas e as funcionalidades do sistema. São feitos fluxogramas dos novos processos, e cada actividade é detalhadamente descrita. Na pormenorização das actividades, inicia-se a configuração do sistema.

Configuração, ou parametrização, pode ser definida como uma preparação do ambiente para implementar os processos da empresa, isto é, de entre todas as opções oferecidas pelo sistema, deve-se, no momento da configuração, escolher e definir campos, parâmetros e modos de executar funcionalidades, de forma a colocar no sistema as regras do negócio. Isto ocorre, visto que os sistemas ERP são produzidos para atender a diversos tipos de empresas e, sendo assim, possuem inúmeros parâmetros que devem ser seleccionados de acordo com o negócio.

Numa fase posterior, são realizados testes de integração. Primeiramente testes funcionais, ou seja, funcionalidades dos módulos e, em seguida, testes entre os vários módulos. Também devem ser realizados testes de interfaces, caso existam, o que ocorre quando outros sistemas ainda são mantidos.

A formação dos utilizadores finais também já deve ser iniciada.

As migrações de dados (ou carregamento de dados) são realizadas em duas fases. Na primeira, antes dos testes de integração, é feita uma migração de dados em pequena escala, testando assim os programas de migração e fornecendo dados para os testes de integração. O carregamento final de dados é iniciado dias antes da entrada em produção do sistema, e contempla a migração, por exemplo, do registo de fornecedores, clientes, lista de materiais, contas a receber, contas a pagar, stocks, entre outros. Para tal, os sistemas antigos devem ser retirados de operação.

A última etapa corresponde à entrada do sistema em produção, e é designada por “Go Live”.

Causas do fracasso / sucesso na implementação de um ERP

A principal causa do fracasso da implementação de um ERP é, sem dúvida, subestimar a reestruturação e mudanças necessárias para a implementação de um ERP na empresa.

Existem ainda outros factores que contribuem para que tal projecto seja um sucesso ou, pelo contrário, para que o projecto fracasse:

- ✓ A escolha do fornecedor / produto ERP adequado;

- ✓ O empenho da direcção;
- ✓ Uma atribuição suficiente de recursos (salas, pessoal interno, consultores, *software* e *hardware*);
- ✓ Um planeamento do projecto cuidado e pormenorizado;
- ✓ A motivação, conhecimento dos processos e a sinergia resultante das equipas envolvidas no projecto;
- ✓ Uma organização do projecto suficientemente estruturada, mas também suficientemente fluida, para que as decisões sejam tomadas rapidamente;
- ✓ Uma modelação cuidada dos processos da empresa;
- ✓ A atenção com a formação dos utilizadores finais porque serão eles a utilizar o sistema;
- ✓ Paciência: a implementação de um ERP é demorada.

2.3 Integração de Sistemas de Informação

2.3.1 Motivações e evolução da integração de SI

Actualmente a integração de SI é um factor crítico na maioria das organizações e é indispensável à agilização e optimização dos processos de negócio.

Silva (2003) realça a necessidade de integrar uma nova aplicação ou SI numa empresa com as demais aplicações existentes e compara-a com a necessidade que um novo empregado tem de relacionar-se com os demais empregados da empresa. Caso não aconteça, a nova aplicação fica isolada das restantes, o que obriga que qualquer actividade seja registada de forma manual. Consequentemente, isto acarreta elevados custos e dá origem à ocorrência de erros.

Favaretto (2006) afirma que as empresas utilizam SI para suporte a decisões. Estes possuem entradas que são transformadas em saídas, através de processamentos internos. Quando as entradas são limitadas a uma actividade da empresa, as saídas também o serão, o que torna o SI isolado e, consequentemente, as decisões tomadas também serão limitadas. Portanto, o mesmo autor defende que, para aumentar a capacidade de decisão, é necessária a existência de um SI abrangente, que deve integrar os diversos subsistemas.

Visto que actualmente as empresas são constantemente pressionadas pela concorrência e pelos seus clientes, a disponibilidade de informações precisas e actualizadas é necessária para suportar boas decisões.

Para Linthicum (2000), citado por Custódio (2006, p. 3) o mundo dos negócios torna-se a cada dia que passa mais complexo e competitivo que nunca, e as empresas que recorrem às aplicações existentes para gerir o negócio, deparam-se com um problema. A escolha das TI a usar era normalmente feita a nível dos departamentos, consoante as necessidades de cada um, o que fez com que dentro de uma organização existissem várias tecnologias independentes e por vezes incompatíveis entre si. A não existência de cruzamento de informação entre sistemas fez com que as empresas sentissem a necessidade de um processo que permitisse interligar toda a organização.

Nos últimos anos, a proliferação de SI nas organizações contribuiu para a identificação e divulgação dos grandes desafios e esforços necessários para a integração dos SI. (De Sordi e Marinho, 2007)

De acordo com Martins (2006), a constante evolução das TI criou realidades tecnológicas díspares, fomentando a necessidade de partilhar informação e funcionalidades entre sistemas.

Segundo Silva (2003), a necessidade de integração de SI tem vindo a aumentar substancialmente e pode ser explicada pelos seguintes factores:

- ✓ Aumento da diversidade e complexidade dos SI nas organizações;
- ✓ A tendência crescente que as empresas demonstram em adquirir SI a fornecedores, ao invés de desenvolver soluções à medida;
- ✓ Alguns SI, principalmente no sector financeiro e administração pública, tornaram-se obsoletos, o que originou o desenvolvimento de novos SI que, em grande parte dos casos, são extensões dos SI existentes e por isso devem estar bem integrados com estes;
- ✓ O surgimento e a evolução das novas tecnologias baseadas na Internet contribuíram para a integração de SI, seja dentro das empresas ou entre empresas;
- ✓ O facto de os ERP não resolverem todos os problemas das empresas, leva a manter outros SI ou desenvolver novos, que devem estar devidamente integrados com o ERP;
- ✓ O reconhecimento dos processos de negócio como chave para desenhar os SI modernos, veio obrigar a redesenhar a arquitectura tecnológica dos SI empresariais, criando novas necessidades de integração.

De Sordi e Marinho (2007) mencionam outros factores, nomeadamente:

- ✓ Procura de vantagens competitivas que requerem melhor gestão da informação;

- ✓ Exigência por parte dos órgãos reguladores por uma maior agilidade no trâmite de informações;
- ✓ Tendência por trabalhos organizados de forma colaborativa, exigindo melhor fluxo informacional entre as organizações.

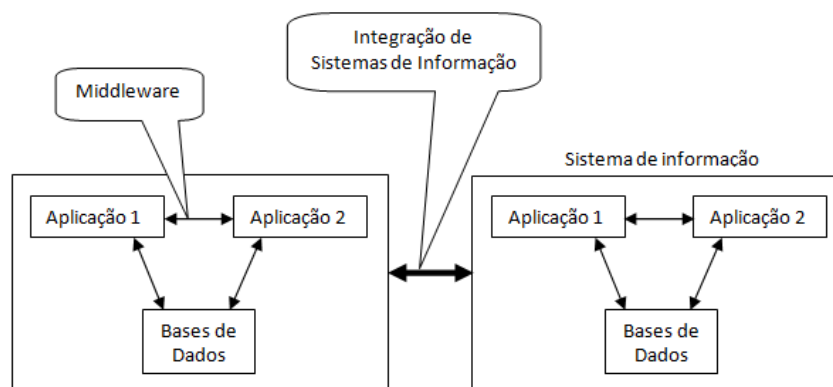


Figura 6: Integração de Sistemas de Informação (Adaptado de Silva, 2003)

Segundo Martins (2006), as soluções para a integração de SI passaram por várias fases durante o seu processo de evolução. Começaram com a criação de vários sistemas interligados, a que se seguiu a adoção de ERP, que entretanto foram complementados com sistemas do tipo *Customer Relationship Management* (CRM), *Supply Chain Management* (SCM) e outras aplicações já existentes na organização. Com a crescente necessidade de integração de SI e com o surgimento de várias aplicações específicas de negócio, foram surgindo soluções específicas, que são denominadas por *Enterprise Application Integration* (EAI).

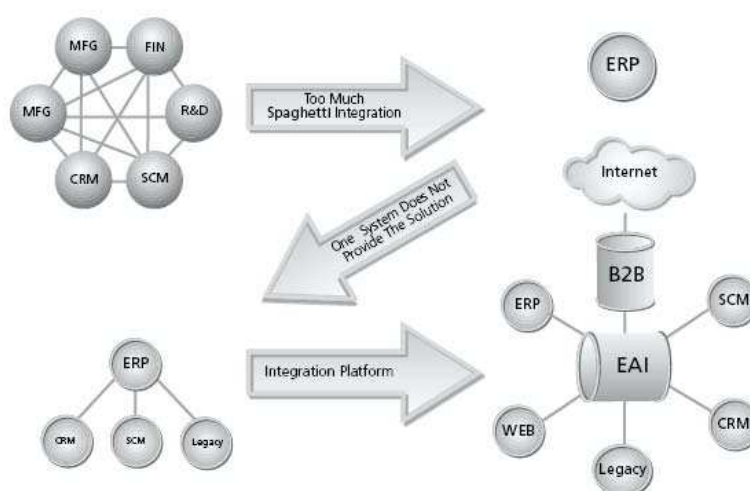


Figura 7: Evolução da integração de SI (Adaptado de Martins, 2006)

2.3.2 Conceito de Enterprise Application Integration (EAI)

Conforme foi referido anteriormente, o crescimento das TI foi um factor chave para o surgimento dos EAI, uma vez que muitas empresas chegaram a um ponto em que tinham várias tecnologias críticas para o negócio mas que não interagiam entre si.

Os EAI têm como objectivo a ligação entre aplicações isoladas, permitindo que as empresas possam concentrar grande parte dos seus esforços na criação de competências em detrimento da gestão de relações entre actividades.

Ruh *et al.* (2001) defendem que o EAI é a criação de novas soluções estratégicas de negócio combinando as funcionalidades das aplicações existentes na empresa e novo código usando *middleware* comum.

Por *middleware* entende-se um tipo de *software* que permite a comunicação entre duas ou mais aplicações.

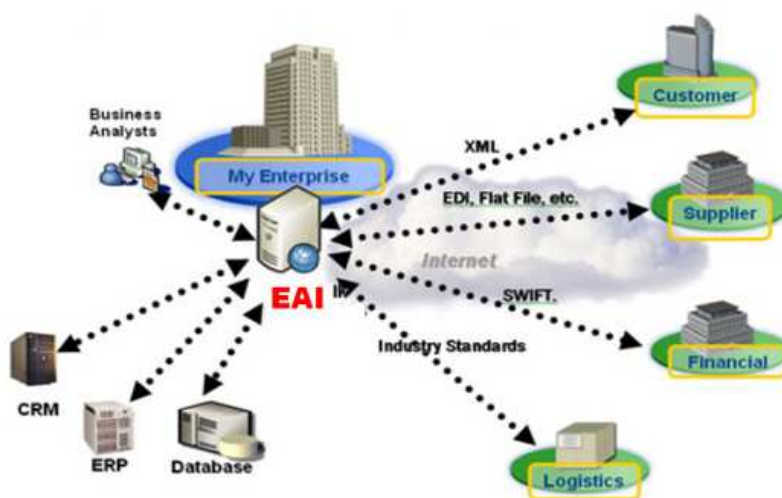


Figura 8: Esquema de integração EAI (Fonte: <http://www.fujitsu.com>)

Ainda segundo os mesmos autores, o EAI melhora a relação da empresa com o cliente, pois este passa a vê-la como uma única entidade, em vez de estar fragmentada por departamentos.

Zaidan (2006) aponta os seguintes factores como preponderantes no sucesso da implementação de um sistema EAI:

- ✓ Visão da integração compartilhada por todas as hierarquias da empresa;
- ✓ Comprometimento com os padrões;

- ✓ Empresas orientadas ao serviço, de modo a concretizar as suas responsabilidades corporativas;
- ✓ Infra-estrutura compartilhada para dar suporte a todas as aplicações;
- ✓ Facilidade no acesso aos dados.

2.3.3 Âmbitos de integração

As tecnologias necessárias para integrar aplicações e SI entre empresas e na própria empresa são basicamente as mesmas, excepto, eventualmente, na segurança que terá de ser garantida entre empresas. Por outro lado, as empresas estão muitas vezes divididas em “unidades de negócio” que são praticamente “empresas dentro da empresa”, com departamentos funcionais próprios.

É também habitual que as empresas aprofundem mais a sua relação com outras de várias formas, tais como parcerias, associações, *holdings*, grupos económicos e redes de *franchising*. É normal que duas ou mais empresas concorram juntas a um projecto e, para esse projecto, se comportem como se tratassem de uma única empresa virtual. (Silva, 2004).

De acordo com Silva (2003), existem várias formas de classificar a integração de SI, mas provavelmente a mais simples e a mais importante é baseada na geografia e pode ser dividida pelos seguintes tipos:

- ✓ **Integração no computador** – ocorre quando os SI partilham a mesma máquina;
- ✓ **Integrações na empresa** – neste tipo de integração, os SI residem em máquinas distintas, mas estão ligadas através de uma rede de pequena latência;
- ✓ **Integração entre empresas** – ocorre quando os SI estão ligados ocasionalmente e/ou quando são geridos por entidades distintas.

Entretanto, o mesmo autor realça que não existem fronteiras claras e bem definidas entre os três tipos de integração acima descritos, devido ao facto de existirem casos específicos como a existência de circuitos dedicados a interligar empresas distintas, o que fará com que o cenário seja semelhante ao de uma única empresa ou, em sentido contrário, existem casos de empresas com várias filiais fisicamente dispersas, o que se assemelha a uma integração entre empresas.

2.3.4 Tipos de integração

Ao longo dos tempos, vários autores tem vindo a apresentar diferentes abordagens à integração de sistemas, que têm evoluído com o avanço da tecnologia.

Para Linthicum (1998), existem quatro tipos de abordagens de integração: ao nível dos Dados, Interface de Aplicação, Métodos e Interface de Utilizador.

Ruh *et al.* (2001) consideram três níveis: Apresentação, Funcional e de Dados.

Segundo Silva (2003), existem quatro níveis de integração: Dados, Lógica (Métodos), GUI (Interfaces com o utilizador), que correspondem à arquitectura tradicional de três camadas e um quarto nível constituído por Portal e Processos, enquadrados numa visão orientada ao negócio.

De seguida passa-se a descrever os três níveis de integração, baseados na arquitectura tradicional referida por Silva (2003) - Dados, Funcional e Interfaces com o utilizador:

Integração via Dados

A integração via dados é o processo (bem como as técnicas e tecnologias) de mover dados entre bases de dados, o que torna possível obter informação de uma base de dados para actualizar outra. Esta abordagem é uma das mais utilizadas pelas empresas que se iniciam no EAI pois o acesso à base de dados é relativamente simples e não é preciso alterar significativamente a estrutura das aplicações e das bases de dados. Uma das vantagens deste modelo de integração é o facto de permitir a comunicação e troca de informação entre as aplicações mais isoladas. (Linthicum, 2000)

Segundo Silva (2004), a integração ao nível dos dados inclui a transformação e a aplicação lógica dos dados de negócio que podem ser extraídos ou lidos (através de *triggers*, *stored procedures*, entre outros).

De acordo com Linthicum (1998), citado por Garcia e Shinotsuka (2001, p. 6), diversas ferramentas têm sido utilizadas para aceder e integrar informações de bases de dados. De seguida, são apresentadas algumas dessas ferramentas:

- ✓ **Batch file transfer** – é considerada com umas das primeiras ferramentas utilizadas para a integração de dados e permitem mover arquivos entre sistemas e aplicações;

- ✓ **Open Database Connectivity (ODBC)** – é uma aplicação padrão de programação de interface que permite aceder a bases de dados heterogéneas, através da inserção de um driver de base de dados entre a aplicação e o Sistema de Gestão de Bases de Dados (SGBD).
- ✓ **Database access middleware** – É um tipo de *software* que proporciona a conectividade entre bases de dados distribuídas. Concentra-se na troca de interrogações, gestão de resultados, conectividade entre bases de dados e outras tarefas de gestão de dados. (Ruh *et al.*, 2001)
- ✓ **Data transformation** – é uma ferramenta que normalmente complementa o *middleware*. Fornece os meios para converter informação da base de dados de origem para a base de dados destino, pois habitualmente as estruturas e esquemas de dados variam entre aplicações. (Ruh *et al.*, 2001)

Segundo Silva (2004), este tipo de integração deve ser utilizado sempre que se pretenda:

- ✓ Obter uma solução moderadamente simples;
- ✓ Combinar dados de múltiplas fontes para análise e tomada de decisões;
- ✓ Permitir acesso de leitura a dados chave a múltiplas aplicações;
- ✓ Extrair, reformatar os dados e actualizar outro sistema.

Este processo tem os seus prós e contras, nomeadamente:

- ✓ Permite o acesso completo ou parcial aos dados, consoante a necessidade da aplicação;
- ✓ Simplifica o acesso às bases de dados, quer através de *middleware* ou de outras ferramentas, o que torna a integração mais rápida e cómoda;
- ✓ Permite a reutilização e partilha de dados entre aplicações, mas para os usar as aplicações têm de ter funcionalidades para manipulação dos mesmos;
- ✓ A integração está dependente dos modelos de dados, logo, se os modelos mudarem a integração pode ficar comprometida. Uma vez que os sistemas tendem a evoluir, pode ser difícil manter a integração.

Para dados sem importância e bases de dados simples, a integração orientada aos objectos pode ser uma opção viável. No entanto, este tipo de integração não deve ser considerado em sistemas críticos, quando os dados são complexos ou quando é necessário que estes sejam sujeitos a

regras complexas de transformação ou processamento antes de serem lidos ou escritos. (Silva, 2003)

Integração Funcional

A integração funcional é a partilha da lógica de negócio. O mecanismo de partilha de métodos pelas aplicações inclui procedimentos remotos, objectos distribuídos, servidores de aplicação ou monitores de transacção. A tecnologia baseada em mensagens é mais incaracterística, no entanto, é também incluída nesta categoria. (Silva, 2004)

Silva (2003) defende que, em termos tecnológicos pode-se considerar o nível funcional como o mais correcto para fazer qualquer integração, até porque o propósito de integrar SI muitas vezes nem sequer é trocar dados, mas sim obter remotamente um serviço prestado por uma aplicação.

Silva (2004) apresenta os seguintes pontos como sendo os mais identificativos para a utilização deste tipo de integração:

- ✓ Integrações para garantir consistência de dados em diferentes aplicações;
- ✓ Integrações de múltiplos passos consecutivos, suportadas por diferentes aplicações (workflow, etc.);
- ✓ Construir aplicações “virtuais” à custa das existentes;
- ✓ Reutilização de tarefas já existentes por novas aplicações.

Alguns autores consideram este modelo como o mais flexível, embora exija um esforço considerável para ser implementado a nível de toda a empresa.

A grande vantagem deste tipo de integração é que permite uma troca de dados síncrona (com pedido, bloqueio e resposta) ao nível aplicacional, exactamente a forma como trabalham localmente as linguagens de programação modernas baseadas em procedimentos, métodos ou componentes. (Silva, 2003)

Integração via Interface de Utilizador

Segundo Silva (2003), a integração de SI orientada às interfaces utiliza as interfaces com o utilizador como ponto de entrada no SI, com o objectivo de simular o comportamento do utilizador tanto para inserir como para obter dados.

Um exemplo cada vez mais actual é a utilização de portais *e-business* na Internet, que podem representar uma simples camada do *front-end* para *Web* de aplicações que já existiam, desenvolvidas sem ter por objectivo a Internet. (Silva, 2004)

De modo a implementar correctamente este modelo de EAI deve-se primeiro compreender a aplicação, a sua organização de dados, o seu esquema lógico e como os dados são apresentados no ecrã. A informação apresentada no ecrã não vem necessariamente da base de dados, podendo haver informações manipuladas por funções lógicas do programa, fazendo com que seja preciso localizar os dados que deram origem a essa informação. Isto faz-se através da leitura da documentação do programa. Uma vez compreendida a aplicação, deve ser criado um catálogo contendo toda a informação passível de aparecer em cada ecrã e também deve ser documentada a informação gerada através de fórmulas ou recebida do utilizador. Para mapear a informação apresentada na interface do utilizador usa-se um processo chamado *screen mapping* que localiza cada elemento de dados em cada ecrã, o seu tamanho e outras informações relevantes. (Custódio, 2006)

Segundo Linthicum (2000) citado por Custódio (2006), a integração pela interface com o utilizador cria uma automação que pretende simular um utilizador a navegar através de ecrãs, carregar em botões e a ler ecrãs, retirando a informação, tratando-a e apresentando-a na interface virtual alvo.

A grande vantagem deste tipo de integração baseada em interfaces é que, tal como na integração baseada em dados, não é necessário lidar com código. Além disso, permite integrar aplicações mesmo quando a própria base de dados não está acessível. Neste caso, todas as validações e restrições de integridade são verificadas, uma vez que associada à interface existe a lógica de integridade que é processada antes dos dados serem lidos ou escritos; o que resulta num elevado índice de segurança e fiabilidade relativamente aos dados. (Silva, 2003)

Como desvantagens podemos apontar o facto de as interfaces dos SI incluírem muita informação, o que dificulta a identificação dos dados relevantes; potenciais problemas de desempenho; a instabilidade, resultante da frequente alteração do desenho das páginas, entre outras.

2.3.5 Tecnologias de integração

Nos dias de hoje, as tecnologias deixaram de ser um obstáculo à integração dos SI. O sucesso dos projectos depende de processos, que são transversais a toda a organização e que devem ser correctamente suportados pela tecnologia. (Marques, 2004)

Com a natural evolução dos recursos tecnológicos, têm surgido novas soluções e novas ferramentas para facilitar a integração de SI.

Neste tópico serão abordadas algumas das principais tecnologias utilizadas para a integração de SI, nomeadamente:

- ✓ *Extended Markup Language* (XML);
- ✓ Monitores Transaccionais (MT);
- ✓ *Remote Procedure Calls* (RPC);
- ✓ *Message Oriented Middleware* (MOM);
- ✓ *Message Brokers*;
- ✓ Objectos distribuídos;
- ✓ *Web Services* e a Arquitectura Orientada aos Serviços;

XML

A integração da informação numa organização ocorre essencialmente ao nível do seu armazenamento e do seu transporte de uma fonte para outra. O XML é destacado como a norma mais abrangente para todo o tipo de manuseamento e integração de informação em diversas soluções tecnológicas. (Martins, 2006)

Cummins (2002), citado por Zaidan (2006, p. 44), justifica o domínio do formato XML na troca de dados entre sistemas, com o facto da utilização de documentos XML fornecer uma melhor flexibilidade e um formato que pode ser trocado facilmente na Internet e filtrado por *firewalls*, usando o protocolo HTTP (*HyperText Transfer Protocol*). Além disso, aponta ainda a existência de funções comuns de *software* disponíveis para a manipulação e a transformação de documentos XML.

Martins (2006) define o XML como uma norma que especifica uma linguagem padrão para a representação de informação independentemente das tecnologias. O XML é uma linguagem de *markup* que consiste num conjunto de regras para formatar e estruturar dados explicando o seu

significado. Um documento XML usa uma sintaxe simples e de fácil compreensão constituído por uma árvore de elementos relacionados e rotulados.

Para Silva (2003), a grande vantagem do XML é a possibilidade de ser utilizado não só para formatar documentos estruturados, mas também estruturas de dados extremamente rígidas. Além disso, pode ser utilizado para estruturar documentos sem uma estrutura definida.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<Aluno>
  <Identificacao>
    <Num_Aluno>23900</Num_Aluno>
    <Nome>Manuel Santos</Nome>
    <Sexo>Masculino</Sexo>
    <DataNascimento>1990/01/10</DataNascimento>
    <Endereco>Rua Renato Cardoso, 38</Endereco>
    <Cidade>Mindelo</Cidade>
  </Identificacao>
</Aluno>
```

Figura 9: Exemplo de um ficheiro XML

Na estrutura de um documento XML, a primeira linha identifica que o documento está no formato XML e indica também a versão. A estrutura hierárquica permite facilmente perceber a forma como a informação está representada. O documento representado na Figura 9, permite representar uma ficha de aluno, onde o mesmo é caracterizado pelo seu número de aluno, nome, sexo, data de nascimento, endereço e cidade.

À semelhança do *HyperText Markup Language* (HTML), o XML permite a definição de *tags* ou delimitadores, que caracterizam os dados e a sua formatação, mas de forma ilimitada. É possível criar um conjunto de *tags* ou marcas específicas para obter uma formatação para um determinado conjunto de informação. Cada elemento, ou conjuntos de elementos, está relacionado e estruturado numa lógica de árvore ou hierarquia com dependências e ramificações. Uma folha da árvore será um elemento que pode conter dados como o título de um livro, o código de um produto ou qualquer outro tipo de dado. Com o XML pode-se criar uma camada estruturante para um documento de texto, um registo de base de dados estruturado, um objecto com métodos e dados como são os objectos Java, um registo de dados resultantes de uma pesquisa, a apresentação gráfica de uma aplicação, etc. (Martins, 2006)

Um XML correctamente estruturado só tem um “*root element*”.

Monitores Transaccionais

De acordo com Martins (2006), os Monitores Transaccionais (MT) ou *Transaction Processing Monitors*, são também designados por *middleware* e correspondem a um ambiente de

execução transaccional de aplicações. Trata-se de uma tecnologia muito popular em ambientes de *mainframes*, porém podem ser utilizados noutras situações já que fornecem um conjunto de serviços para sistemas num ambiente distribuído como é o cliente/servidor. Estes serviços vão desde o controlo do acesso às bases de dados, controlo de *workflow*, gestão de transacções entre aplicações e até suportam as comunicações com a interface do utilizador.

Uma transacção pode ser definida como uma acção que provoca uma mudança de estado de uma organização. Um exemplo de uma transacção poderá ser uma transferência bancária de uma conta para outra, executada por um cliente.

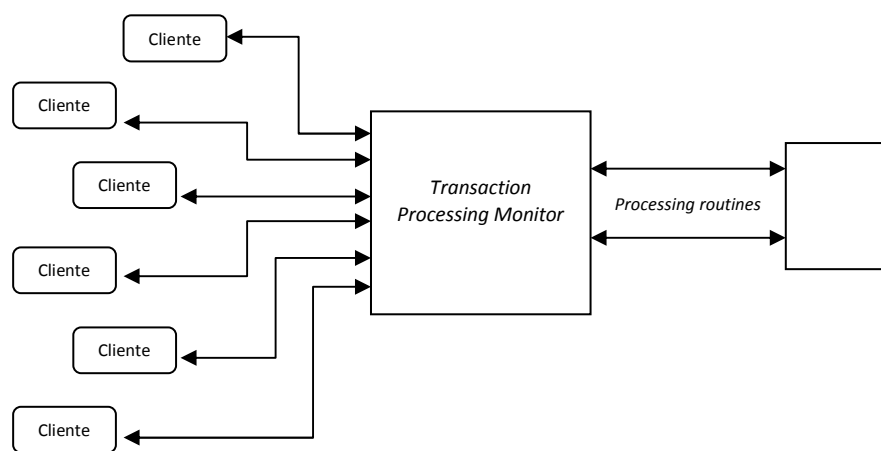


Figura 10: Monitor Transaccional (Adaptado de Martins, 2006)

Para Martins (2006), uma transacção é composta por um conjunto de operações atómicas e só tem sucesso caso todas as operações sejam executadas com sucesso. Este conceito é aplicado em muitas tecnologias como por exemplo as bases de dados transaccionais ou em motores de *workflow*. Um monitor transaccional fornece um conjunto de funções como por exemplo a gestão de transacções falhadas, o balanceamento de carga, a consistência e destruição de informação e níveis de segurança na execução das transacções.

Uma transacção é gerida desde a sua origem, em geral o cliente, ao longo de múltiplos servidores, até à sua conclusão, regressando o controlo à sua origem.

Remote Procedure Call (RPC)

A tecnologia de integração baseada no conceito RPC permite que uma aplicação possa chamar procedimentos localizados noutras máquinas. Um procedimento pode ser definido como um conjunto de código que faz parte de um programa, implementa um serviço bem definido e

oferece uma interface para que outras partes do mesmo programa possam pedir a execução desse serviço.

Martins (2006), define o RPC, no contexto de uma arquitectura cliente/servidor, como um protocolo que permite a um programa executar outro programa remoto e armazenado num servidor, em que o programa cliente envia uma mensagem para o servidor com os respectivos parâmetros e o servidor responde com uma mensagem com a informação resultante da execução.

A linguagem IDL - *Interface Definition Language*, especifica as características do procedimento fornecido pelo servidor ao seu cliente. O código de chamadas à rede é escondido em procedimentos chamados *stubs*, gerados automaticamente por um compilador (exemplo: o RPCGen da Sun), abstraindo as aplicações (clientes e servidoras) de preocupações com detalhes, tais como *sockets*, poupando muito esforço de programação. (Silva, 2004)

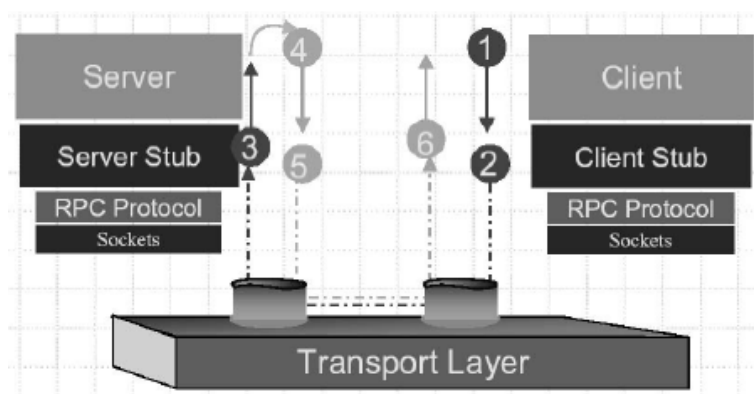


Figura 11: Remote Procedure Call

Message Oriented Middleware (MOM)

O MOM foi criado com o objectivo de estabelecer um padrão para *middleware* de mensagens. MOM permite que mensagens sejam trocadas em ambientes cliente/servidor através de filas de mensagens.

O MOM é uma infra-estrutura de *software* cliente/servidor que cria uma camada entre as aplicações de alto nível e as plataformas onde estão instaladas, substituindo a comunicação directa entre aplicações por um sistema de troca de mensagens.

De acordo com Martins (2006), o MOM permite um elevado nível de flexibilidade e independência para a troca de mensagens entre aplicações e sua consequente integração. Este tipo de arquitectura é indicado para aplicações que necessitam de uma integração orientada a eventos.

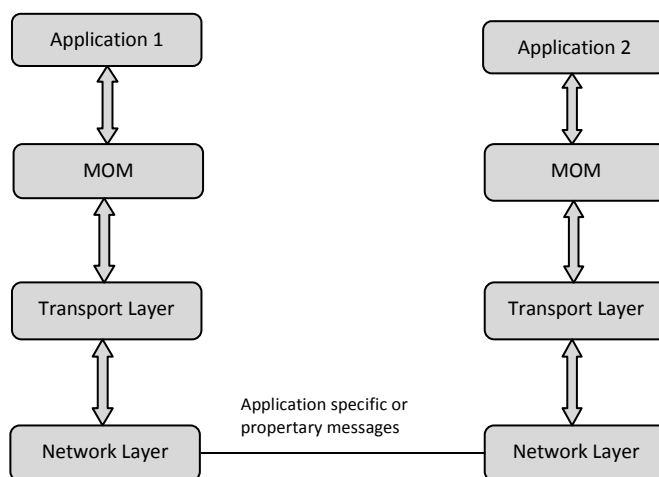


Figura 12: Descrição geral da arquitectura MOM

Message Brokers

Os *message brokers* baseiam-se numa comunicação assíncrona, ou seja, através de um intermediário, entre aplicações e SI. Actuam como peça central de um sistema de integração, sendo constituídos por um conjunto de serviços nucleares que permitem aliar a capacidade de tratar mensagens com a capacidade de *routing*. As suas principais responsabilidades verificam-se ao nível do *routing* de mensagens, transformação de dados, implementação de regras de negócio e suporte aos processos de negócio.

Os *message brokers* funcionam como uma ponte integradora entre aplicações, permitindo a circulação e troca assíncrona de mensagens. Para além da circulação e do encaminhamento das mensagens, os *message brokers* podem ler o conteúdo delas, decodificá-las, interpretá-las, modificá-las e voltar a codificá-las para depois serem entregues no destino. (Martins, 2006)

Por outras palavras, os *message brokers* actuam como intermediário e garantem a segurança e a escalabilidade do sistema, com a vantagem de cada aplicação ter apenas de estar ligada a um canal único de comunicação, em vez de estar directamente ligada a todas as outras aplicações. Este cenário proporciona flexibilidade na integração das aplicações e garante facilidade na configuração do envio e recepção de mensagens entre aplicações.

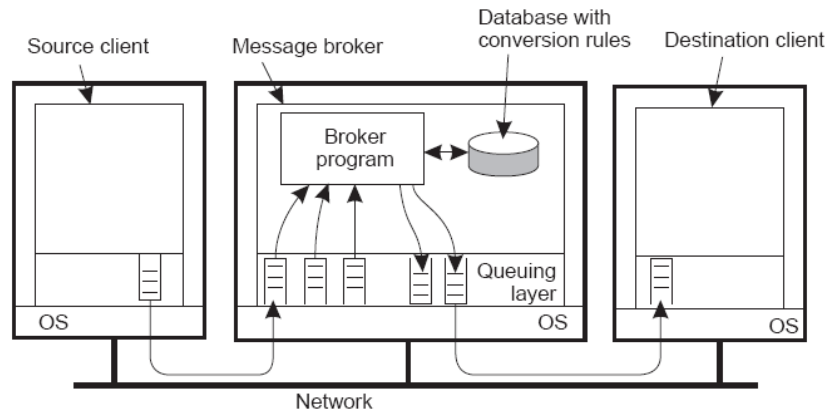


Figura 13: Arquitetura de um *Message Broker*

Objectos distribuídos

O conceito ORB (*Object Request Broker*) corresponde a tecnologia ao nível do *middleware* que gere a comunicação e troca de informação entre objectos distribuídos. Um ORB é um componente de acordo com o modelo CORBA (*Common Object Request Broker Architecture*) do consórcio OMG (*Object Management Group*) que gere a integração de objectos num modelo cliente/servidor. (Martins, 2006)

Os objectos distribuídos surgiram da união das tecnologias de Sistemas Distribuídos e Orientação a Objectos.

Dos vários modelos de objectos distribuídos disponíveis no mercado, destacam-se o Microsoft COM e o CORBA.

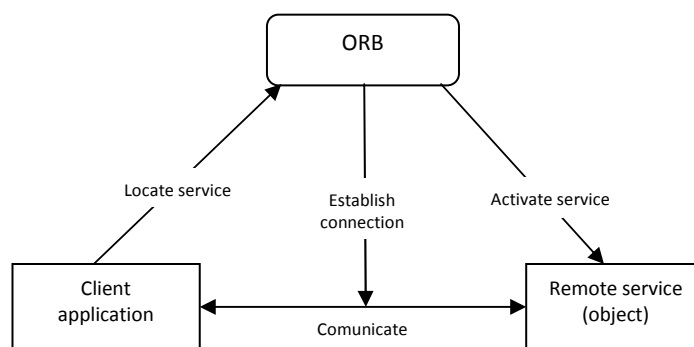


Figura 14: *Object Request Broker* (Adaptado de Martins, 2006)

Segundo Mainetti Jr. (1997), a Microsoft promoveu o OLE (*Object Linking and Embedding*) como uma das suas principais tecnologias para desenvolvimento e integração de aplicações. A

camada base para o OLE é conhecida como COM (*Component Object Model*). Com o lançamento do Windows NT 4.0, no início de 1997, a Microsoft introduziu o DCOM (*Distributed Component Object Model*) que é uma implementação do COM, contemplando características distribuídas de aplicações desenvolvidas numa rede de computadores baseada na plataforma Windows.

Com a implementação do COM foi possível, por exemplo, integrar uma folha de cálculo dentro de uma folha de um processador de texto e manter esse objecto com as características que possuía na aplicação que lhe deu origem.

O padrão CORBA é descrito pela OMG como um padrão que define a interoperabilidade dos objectos num ambiente distribuído. Uma das características mais importantes deste padrão é a existência de uma linguagem para a definição de interfaces IDL, que é uma linguagem também padronizada pela OMG, independente de arquitectura e que possibilita a especificação das interfaces dos objectos distribuídos.

De acordo com Martins (2006), o CORBA versão 2.0 incluiu uma especificação mais clara do canal de comunicação ORB denominado de IIOP (*Internet Inter-Orb Protocol*). O IIOP é baseado nos protocolos da Internet, nomeadamente na norma TCP/IP e permite estruturar a forma como a informação circula entre os objectos.

Segundo Wollrath e Waldo (2005), citados por Martins (2006, p. 126), existe um modelo ORB mais recente, baseado na tecnologia Java, denominado de RMI (*Remote Method Invocation*), que permite a execução remota de métodos de objectos Java. O Java RMI é mais simples que o COM/DCOM ou o CORBA, mas depende do Java. Por seu lado o COM/DCOM depende do Windows e o CORBA não é tão flexível quanto os restantes apesar de estar implementado em várias linguagens.

Web Services e a Arquitectura Orientada aos Serviços (SOA)

De acordo com Cerami (2002), citado por Martins (2006, p. 49), actualmente a integração de sistemas é cada vez mais orientada para as organizações e os seus processos. Com a evolução tecnológica, hoje é possível disponibilizar informação ou procedimentos como serviços, através de uma camada de abstracção suportada por tecnologia do tipo *Web Services* (WS). Os WS são componentes de *software* que permitem computação distribuída baseada em normas da Internet.

Segundo Martins (2006), os WS representam um conjunto estruturado de definições para a integração de diferentes tecnologias, desde o nível dos sistemas operativos, até aos processos organizacionais. Compatibilizam tecnologias e soluções não integráveis à partida pela sua natureza, utilizando formatos comuns de troca de informação com o XML e aplicando regras de interacção. É assim possível evitar tarefas de reconstrução de sistemas obtendo a sua reutilização e respectiva integração.

A utilização de um padrão universal como o XML, para descrever e utilizar os WS, faz com que esta modalidade para integração de sistemas se torne mais prática, fácil e viável, pois é independente de qualquer linguagem ou plataforma.

De acordo com Silva (2003) e Martins (2006), os WS representam um conjunto de normas que definem como descrever, publicar e utilizar determinado tipo de componentes. Essas normas são:

- ✓ **SOAP** (*Simple Object Application Protocol*) - um mecanismo (do tipo RPC) para invocar métodos sincronamente em componentes remotos;
- ✓ **WSDL** (*Web Service Description Language*) - uma linguagem (do tipo IDL do CORBA, mas agora escrita em XML) para descrever formalmente a interface de um serviço que é basicamente um conjunto de métodos agrupados num componente;
- ✓ **UDDI** (*Universal Definition Discovery Interface*) - um WS predefinido que armazena as interfaces de outros WS num repositório e disponibiliza essas interfaces publicamente.

Tecnicamente, os WS são serviços distribuídos que processam mensagens SOAP, codificadas em XML, enviadas através de HTTP e que são descritas através de WSDL.

Silva (2003) define um WS como um componente de *software* que pode ser descrito, publicado, descoberto, invocado e composto.

Os *Web Services* são serviços disponibilizados por um servidor e uma rede interna (Intranet), como nas empresas, ou externa (Internet). Esses serviços são componentes distribuídos utilizados como componentes de aplicações baseadas na *Web*. (Marques, 2005)

Para o correcto funcionamento dos *Web Services* é necessário que algumas tarefas sejam executadas, que podem ser assim resumidas:

- ✓ Criação do *Web Service*, com as suas interfaces e métodos a serem invocados;

- ✓ Publicação do *Web Service*;
- ✓ Localização do *Web Service*.

A figura seguinte ilustra o funcionamento de um *Web Service*:



Figura 15: Funcionamento de um *Web Service* (Fonte: <http://www.sinfic.pt>)

O processo inicia-se com a criação do serviço, actividade na qual o Fornecedor do Serviço é encarregado. De seguida, ocorre a criação do documento WSDL que descreve o novo serviço no formato XML. Após a criação deste documento, o serviço é então publicado num Intermediário de Serviços que, por sua vez, é a entidade responsável pela manutenção e armazenamento de registos de serviços disponíveis e das suas respectivas localizações (UDDI). Surge então, uma terceira entidade denominada por Solicitador ou Cliente do Serviço, que utilizará os serviços mas, antes disso, solicitará a localização de determinado serviço ao Intermediário de Serviços. Este será responsável por efectuar a ligação entre o serviço *web* requerido e o Solicitador do Serviço.

O documento WSDL é um elemento fundamental para que o processo de interacção entre os serviços *web* se inicie. É neste documento onde são descritas informações como a localização, o protocolo de rede utilizado para a comunicação, a definição dos métodos e os tipos de dados e estruturas utilizadas pelo serviço. Este documento pode ser publicado de diversas formas, seja através de um UDDI, que mantém um registo central de serviços, ou submetido ao e-mail do Solicitador do Serviço, por exemplo. O objectivo de todo este processo é fazer com que o Solicitador do Serviço tenha acesso ao WSDL dos serviços disponibilizados pelo Fornecedor do Serviço, para que o vínculo seja efectivado. O próximo passo deste processo é iniciar a invocação remota de métodos através de um protocolo de troca de mensagens em XML, feita através do SOAP (Carvalho, 2003).

Na estrutura de um *web service*, é possível identificar as camadas básicas para o seu funcionamento. A figura seguinte ilustra essas camadas na forma de pilha, onde é possível identificar as tecnologias que são utilizadas em cada camada da pilha:

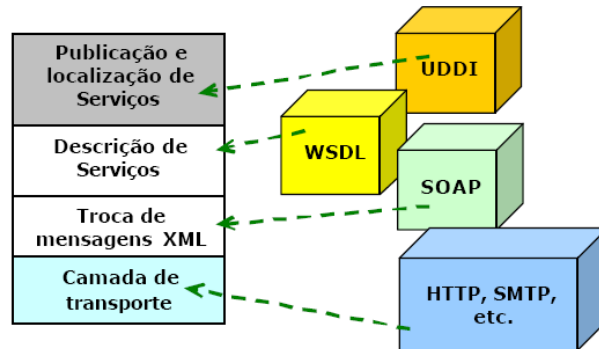


Figura 16: *Web Services Stack* (Adaptado de Carvalho, 2003)

- ✓ **Camada de transporte:** é a camada que disponibiliza o serviço ao meio externo de comunicação. Pode utilizar qualquer um dos protocolos de comunicação, sendo o protocolo http, o mais utilizado;
- ✓ **Troca de mensagens XML:** é a camada responsável pela definição do formato da mensagem que será utilizada na interação entre as aplicações. O padrão que normalmente se utiliza nesta camada é o protocolo SOAP;
- ✓ **Descrição de serviços:** é a camada responsável pelo fornecimento de um mecanismo de descrição da funcionalidade proporcionada pelo *Web Service*. Nesta camada, pode -se encontrar o padrão de descrição de *Web Services*, o WSDL, que segue a especificação XML para descrever o serviço;
- ✓ **Publicação e localização de serviços:** nesta camada reside um mecanismo que armazena, de forma organizada, os registos de localização dos *Web Services*. Através desta camada, os serviços são publicados e os clientes encontram os serviços que procuram, antes de utilizá-los. Uma tecnologia padrão para esta camada é o UDDI.

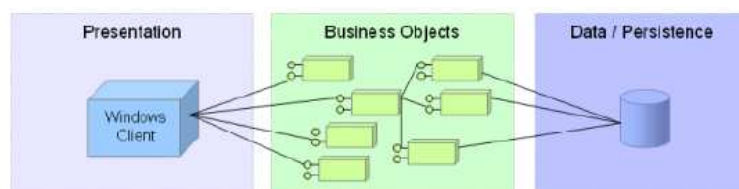


Figura 17: *Arquitectura de 3 camadas* (Adaptado de Martins, 2006)

Segundo Newcomer e Lomow (2004), citados por Martins (2006, p. 100), uma integração assente em *web services* denomina-se por *Service Oriented Integration* (SOI) e, neste âmbito, é possível definir uma arquitectura orientada aos serviços - *Service Oriented Architecture* (SOA). Pode-se considerar que o SOA é uma evolução da arquitectura de 3 camadas (Figura 17), com a incorporação de serviços que disponibilizam e integram as aplicações de acordo com as suas especificações. (Figura 18)

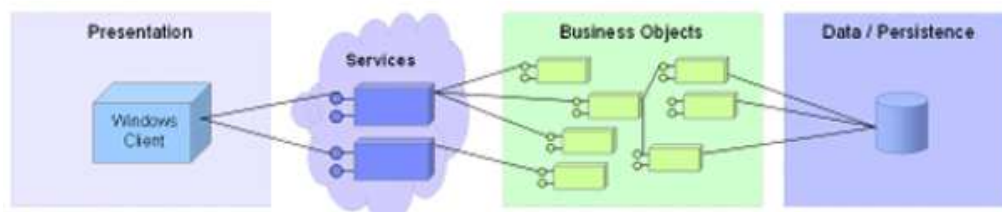


Figura 18: Arquitectura SOA (Adaptado de Martins, 2006)

A arquitectura SOA permite às organizações disponibilizar as suas aplicações e soluções de *software* como serviços bem definidos, permitindo a sua reutilização e integrando-os em soluções mais abrangentes e flexíveis para o desempenho do negócio. Uma arquitectura SOA disponibiliza um repositório de serviços que podem ou não ser integrados numa solução aplicacional. Cada serviço representa um encapsulamento de componentes aplicacionais que contêm lógica aplicacional que segue as regras do negócio. (Martins, 2006)

2.4 Caso de Estudo: Integração entre SAP e Amadeus

Quando se fala de um sistema ERP, um dos primeiros nomes que nos vêm à cabeça é o SAP (Sistemas, Aplicações e Produtos para Processamento de Dados), líder mundial de soluções de *software* empresarial.

Esta secção tem a finalidade de descrever as principais características e a arquitectura do ERP da SAP, bem como de apresentar, como caso de estudo, uma integração envolvendo os sistemas SAP ERP e a aplicação de gestão de viagens da Amadeus, líder mundial em Sistemas de Distribuição Global (GDS - *Global Distribution System*) e no fornecimento de soluções tecnológicas para a indústria de viagens e turismo.

2.4.1 Características e arquitectura do Sistema Integrado SAP R/3

Em 1992, a empresa SAP AG lançou o sistema R/3, solução baseada num modelo de arquitectura cliente/servidor. O sistema R/3 é um *software standard* integrado, especificamente desenhado para funcionar em arquitecturas de sistemas abertos.

SAP é um pacote do tipo ERP, baseado no planeamento de recursos corporativos, através de sistemas integrados de gestão implementados por *software*, constituído pelos seguintes módulos (Santos *et al.*, 2003):

- ✓ **AM** – *Asset Management* (Gestão do Património): faz a gestão de todo o património da empresa, avaliando a evolução da apreciação ou depreciação do mesmo;
- ✓ **PS** – *Project Systems* (Gestão de Projectos): serve para gerir os projectos da empresa;
- ✓ **HR** – *Human Resources* (Recursos Humanos): é utilizado para processamento de salários, formação, gestão de carreiras, etc;
- ✓ **PM** – *Plant Maintenance* (Manutenção Fabril): Utiliza-se para manutenção de todo o equipamento, incluindo trabalho e materiais;
- ✓ **MM** – *Materials Management* (Gestão de Materiais): coordena o aprovisionamento e disponibilidade de materiais, incluindo requisições e ordens de compra, a recepção de mercadorias, as existências, as listas de materiais (BOMs), etc;
- ✓ **QM** – *Quality Management* (Gestão da Qualidade): planeamento, execução e gestão das inspecções e dos certificados;
- ✓ **PP** – *Production Planning* (Planeamento de Produção): gere a produção, contendo o plano mestre de produção, o *shop floor*, o MRP e o planeamento das capacidades;
- ✓ **SD** – *Sales and Distribution* (Vendas e Distribuição): este módulo serve para organizar as vendas, contendo as ordens de venda, o processo de *picking*, a expedição e a facturação;
- ✓ **WF** – *Workflow* (Fluxo de Trabalho): Liga os módulos SAP com outras aplicações, ferramentas e serviços;
- ✓ **IS** – *Industry Solutions* (Soluções Industriais): Este módulo é um módulo complementar desenvolvido especificamente para indústrias tais como petróleo e gás, farmacêutica, etc;
- ✓ **FI** – *Financial Accounting* (Contabilidade Financeira);
- ✓ **CO** – *Controlling* (Contabilidade Analítica).



Figura 19: Módulos do SAP R/3

O SAP R/3 integra uma organização de processos de negócios através da cadeia de fornecimento – desde o fornecedor até ao cliente – dentro de uma rede virtual de comunicação de informação. O sistema é modular, extensível, aberto e flexível e permite que as empresas o adaptem às suas necessidades.

As aplicações e operações de processos estão integradas e comunicam entre as áreas de negócio, o que possibilita a actualização dos dados. Um sistema SAP/R3 disponibiliza, por exemplo, a visualização de um documento de facturação e contabilidade a qualquer momento, com valores actualizados para um determinado cliente (William, 2008).

A grande vantagem que se obtém com a implementação deste sistema, é a transformação dos processos estáticos em redes dinâmicas entre diferentes entidades (clientes, fornecedores, accionistas), o que contribui largamente para a melhoria da produtividade dos colaboradores.

Uma das principais características do sistema R/3 é a sua capacidade de reconfiguração (mapeamento), de forma a satisfazer as necessidades específicas de um determinado negócio. Este mapeamento é conseguido através da parametrização ou adaptação do sistema às especificidades do negócio.

O processo de configuração é geralmente longo e dispendioso, não só devido à necessidade em compreender totalmente os processos e encontrar uma solução que se ajuste àqueles requisitos, como também pelo facto de haver a necessidade de se parametrizar toda a solução,

tendo em conta as melhores práticas empresariais, padrões internacionais, exigindo uma reengenharia dos processos de negócio (William, 2008).

A arquitectura cliente/servidor de três níveis, está estruturada da seguinte forma (Santos *et al.*, 2003):

- ✓ Servidor de Apresentação (*Presentation Server*);
- ✓ Servidor de Aplicação (*Application Server*);
- ✓ Servidor de Base de Dados (*Database Server*).

Servidor de Apresentação

Tipicamente opta-se por uma instalação do GUI (*Graphical User Interface*) do SAP e outras aplicações de *frontend* directamente nos computadores pessoais (PCs) dos utilizadores (um por utilizador). Isto garante que os recursos, neste nível, estejam disponíveis para cada utilizador e nenhum destes pode ser afectado pelo comportamento de outro.

Servidor de Aplicação

Representa o servidor de informação para os clientes, isto é, responde aos pedidos de serviços efectuados pelos clientes.

Servidor de Base de Dados

Corresponde ao servidor onde reside fisicamente a informação definida e gerida a partir do sistema R/3. Ao contrário dos servidores de aplicação, não é fácil a coexistência de vários servidores de base de dados devido a questões de consistência e sincronismo. Para resolver esta situação deve haver uma programação cliente/servidor eficiente, de forma a libertar o servidor de base de dados de trabalho desnecessário. O esforço deste na salvaguarda e acesso à informação deve ser minimizado.

2.4.2 SAP Netweaver

O SAP Netweaver é descrito pela SAP com uma plataforma de arquitectura de TI flexível, que facilita a implementação e execução de novos processos e estratégias de negócio. Foi desenvolvida com o intuito de promover a integração de aplicações dentro de uma organização, permitindo que sistemas homogéneos e/ou heterogéneos pudessem comunicar entre si.

O SAP Netweaver³ apresenta as seguintes características:

- ✓ Fornece conteúdos de negócio pré-definidos que facilitam a integração;
- ✓ Reduz a necessidade de configuração, permitindo acelerar a implementação;
- ✓ Inclui conteúdos pré-configurados de portal e perfis que melhoram a integração dos indivíduos;
- ✓ Disponibiliza relatórios e análises que visam melhorar a integração da informação, bem como interfaces, a fim de produzirem processos de negócio conjuntos;
- ✓ Disponibiliza ferramentas analíticas e suporte que permitem reduzir custos e aumentar a colaboração;
- ✓ Os conteúdos de negócio são pré-configurados para uma utilização imediata, quer através de *Business Packages* quer como parte dos componentes do SAP Netweaver;
- ✓ Os perfis internos e externos são adequados à empresa, de modo a facilitar a interacção com os seus fornecedores, parceiros e clientes;
- ✓ Baseia-se nos mais recentes modelos e tecnologias da Internet como HTTP, XML e *Web Services*. Por sua vez, assegura a abertura e a interoperabilidade com o Microsoft.NET e ambientes J2EE (*Java 2 Platform Enterprise Edition*), como o *IBM WebSphere*;
- ✓ Permite reunir informação estruturada, não estruturada e harmonizar dados formalmente retidos em vários sistemas, através de conteúdos de negócio pré-definidos;
- ✓ É uma plataforma aplicacional e de integração que suporta as SAP xApps e todas as soluções *mySAP Business Suite*; Paralelamente, materializa uma nova arquitectura aplicacional, denominada *Enterprise Services Architecture*, que combina aplicações empresariais com *Web Services* e tecnologias abertas, permitindo a implementação de processos de negócio adaptáveis.

2.4.3 Âmbito da integração entre o SAP Travel Management e o Amadeus e-Travel Management

Conforme referido anteriormente, o objectivo desta secção consiste na apresentação de um caso de sucesso, baseado numa integração entre dois sistemas, como consequência da parceria estabelecida entre a SAP e a Amadeus.

³ <http://www.sap.com/platform/netweaver/pdf/BWP_OV_SAP_NetWeaver.pdf> Acesso em 10 de Setembro de 2010.

O principal objectivo de um sistema de gestão de viagens corporativas é a possibilidade de ter um repositório centralizado de viagens que monitoriza o gasto em viagens e representações. As empresas implementam a gestão dos programas de viagens para ter um maior controlo sobre as despesas de viagens e para gerir os seus viajantes de uma maneira mais eficaz. As despesas com viagens são um dos maiores factores de custos com os recursos humanos e oferecem um grande potencial para a redução de custos.

O SAP Travel Management é um módulo do ERP da SAP que contribui significativamente para a redução dos custos com os processos de viagens, pois possibilita a redução das instâncias envolvidas no processo de gestão e de reservas de uma viagem, bem como a simplificação do processo em si e a inclusão automática das políticas de viagem específicas da empresa.

O Amadeus e-Travel Management é uma aplicação que auxilia as empresas na gestão dos seus programas de viagens globais com mais eficiência e possibilita otimizar a relação custo/benefício. Este sistema fornece ao utilizador o acesso a milhares de hotéis, voos, ofertas de aluguer de viaturas, entre outros.

Grandes organizações como a Bosch, a Nike, a Phillip Morris, a Siemens, a Nestlé, entre outras, adoptaram a solução integrada SAP Travel Management com Amadeus e-Travel Management.

Segundo um estudo realizado pela Bosch, para organizações com várias filiais espalhadas pelo mundo, a centralização da gestão de viagens num único ponto aumenta o poder de negociação com as companhias aéreas, redes de hotéis e outras empresas do ramo, além de facilitar a definição e o cumprimento de políticas corporativas.

Nesse mesmo estudo, a Bosch afirma que o custo com o processamento de reservas de viagens sofreu uma redução de 50%, enquanto os custos directos com viagens – especialmente viagens aéreas – foram reduzidos em cerca de 5%.

Esta solução veio contribuir para facilitar a integração dos procedimentos de reservas de viagens e do controlo de políticas numa estrutura de ERP, uma vez que permite combinar o planeamento e a política de viagens com os sistemas financeiro, de compras, pagamentos e recursos humanos.

A integração do SAP Travel Management com o Amadeus e-Travel Management envolve os seguintes elementos na comunicação entre os dois sistemas⁴:

- ✓ Sistema R/3;
- ✓ *Gateway* SAP;
- ✓ *Gateway* Amadeus;
- ✓ Servidor de API⁵ (*Application Programming Interface*);
- ✓ Sistema Central Amadeus (*Amadeus Practice Training* ou *Amadeus Production*).

A arquitectura do sistema é a seguinte:

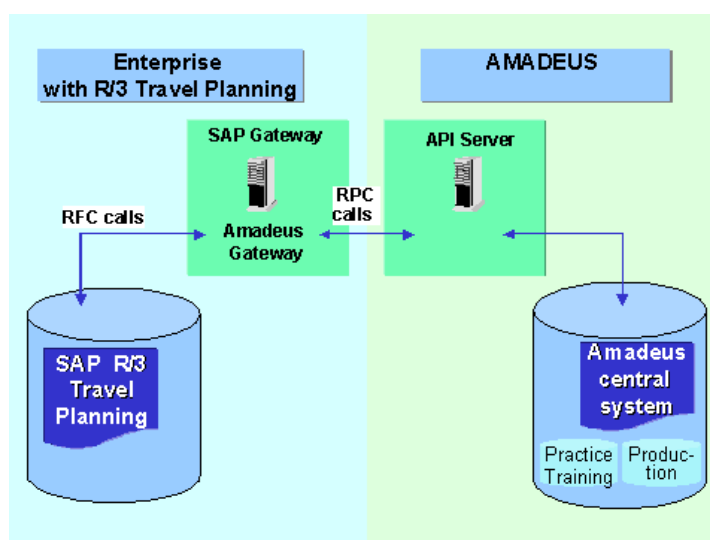


Figura 20: Arquitectura do sistema integrado de gestão de viagens (Fonte: SAP Travel Management (FI-TV))

O destino RFC (*Remote Function Call*) representa a conexão lógica entre o sistema R/3 e o *Gateway* Amadeus. As chamadas RFC são encaminhadas por meio do *Gateway* SAP e este, por sua vez, comunica com o *Gateway* Amadeus. Na arquitectura cliente/servidor, o sistema R/3 assume a função do cliente e o *Gateway* Amadeus assume a função do servidor.

A tarefa do *Gateway* Amadeus é a identificação no servidor de API do Amadeus. O servidor de API controla o acesso ao respectivo sistema central Amadeus (*Amadeus Practice Training* ou *Amadeus Production*). A comunicação entre o *Gateway* Amadeus e o servidor de API do Amadeus é executada por meio de chamadas RPC (*Remote Procedure Calls*). Neste ponto, o *Gateway* Amadeus assume a função do cliente e o servidor de API do Amadeus assume a função do servidor.

⁴ [SAP 2001] Travel Management (FI-TV) – Release 4.6C, 2001.

⁵ Conjunto de rotinas e padrões estabelecidos por um software para a utilização das suas funcionalidades por programas aplicativos que não querem envolver-se em detalhes da implementação do software, mas apenas utilizar os seus serviços.

O *Amadeus Central System* consiste nos seguintes *mainframes*:

- ✓ ***Amadeus Practice Training*** – trata-se de um sistema isolado usado para testes. Os dados deste sistema não são transferidos para os fornecedores externos de serviços. Contudo, esses dados são idênticos aos dados do sistema de produção;
- ✓ ***Amadeus Production*** – é o sistema de produção da solução de viagens do Amadeus, por meio do qual as reservas são encaminhadas directamente aos fornecedores. No sistema *Amadeus Production*, são criados os códigos PNR (*Passenger Name Record*), compostos pelo nome, segmento de voo e endereço do passageiro.

3 Ferramentas

Neste capítulo serão abordadas as ferramentas utilizadas para a integração entre o ERP Primavera e o OpenERP, com enfoque nas suas funcionalidades e motores associados.

3.1 Primavera Business Software Solutions

3.1.1 Características e Arquitectura do ERP Primavera

A Primavera BSS (*Business Software Solutions*) é uma empresa portuguesa vocacionada para o desenvolvimento e comercialização de *software* de gestão e plataformas de integração de processos empresariais para pequenas, médias e grandes organizações e para a Administração Pública, disponibilizando aos seus clientes soluções de Contabilidade, Recursos Humanos, Gestão Comercial, CRM, *Business Intelligence*, *Enterprise Portals*, *Mobile Sales*, assim como softwares específicos para os sectores da Construção Civil e Obras Públicas ou Indústria.

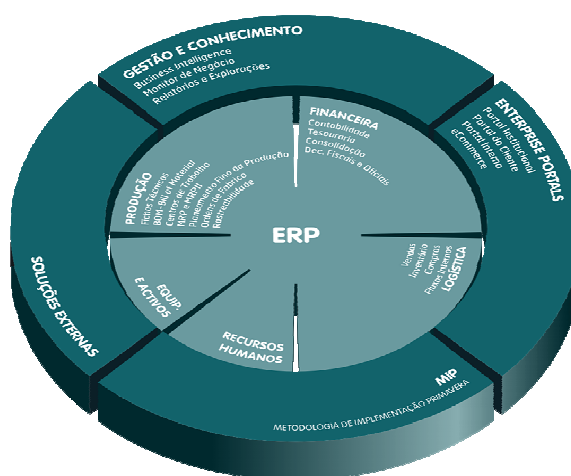


Figura 21: Esquema da Solução Global Primavera (Fonte: Primavera BSS)

Conceito de Extensibilidade

A Extensibilidade das aplicações é um conceito central no ERP Primavera, na medida em que permite complementar a solução *standard* com funcionalidades adicionais e intervir nas funcionalidades nucleares de cada aplicação para responder às necessidades específicas de cada negócio ou cliente em particular.

O ambiente de desenvolvimento utilizado no ERP Primavera é o Microsoft Visual Studio 6 e o sistema de gestão da base de dados é o Microsoft SQL Server.

À medida que os produtos Primavera foram evoluindo, várias tecnologias foram sendo desenvolvidas e exploradas, sendo disponibilizadas aos programadores e implementadores das soluções Primavera. Essas tecnologias apresentam a particularidade de permitirem a reutilização das funcionalidades do ERP a partir de aplicações externas, a reutilização de código e das funcionalidades mais comuns que o ERP disponibiliza, a automatização de processos de negócio específicos, a integração de soluções e aplicações externas no ERP, entre outras vantagens. (Primavera BSS, 2007)

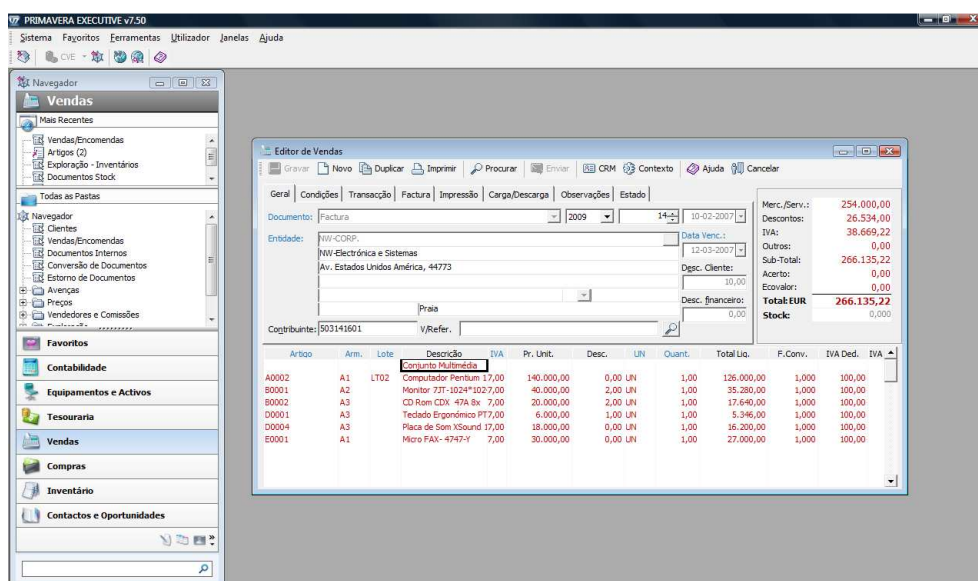


Figura 22: ERP Primavera - Ambiente de trabalho

Tecnologia COM e Arquitectura Windows DNA

Todas as aplicações Primavera são desenvolvidas com base na tecnologia COM (*Component Object Model*) e na arquitectura Windows DNA (*Distributed InterNet Applications Architecture*).

A tecnologia COM é usada para estabelecer a comunicação entre os diversos componentes. O COM é um sistema independente da plataforma de utilização, que permite desenvolver objectos (componentes binários) que podem interagir com outros objectos. Podendo esses objectos estar a correr no mesmo processo, num outro processo, e, até, numa máquina remota. Os objectos COM podem ser entendidos, como disponibilizando: quer um conjunto de dados, quer um conjunto de funções relacionadas de acesso e manipulação desses mesmos dados. A estes conjuntos de funções chama-se *Interfaces*, e às funções pertencentes à interface chama-se métodos. (Matos, 2004)

O *Windows DNA* é um modelo criado pela Microsoft, para desenvolvimento de aplicações para a plataforma *Windows*. Este modelo baseia-se numa arquitectura lógica de três camadas, utilizando componentes reutilizáveis.

Como o próprio nome indica, o modelo de 3 Camadas (*3-Tier Application Model*) separa uma aplicação em três camadas, com funções perfeitamente definidas, isoladas em componentes diferentes (Figura 23):

- ✓ A primeira camada, denominada de Interface do Utilizador, define as Entidades de Negócio (*Business Entities*), como por exemplo, a Ficha de um Artigo, a Ficha de um Cliente, ou um Documento de Venda.
- ✓ A segunda camada – Serviços de Negócio (*Business Services*) – define as Regras do Negócio. Todas as Regras de Negócio da aplicação são da responsabilidade desta camada. Por exemplo, a moeda definida na ficha de determinado cliente, deve ser a moeda utilizada nos documentos de venda desse cliente.
- ✓ Serviços de Dados (*Data Services*) – representam a terceira e última camada e é responsável por todo o acesso à base de dados.

A utilização desta tecnologia permite que diferentes objectos sejam partilhados por diferentes aplicações dentro do ERP Primavera. Por exemplo, é possível aceder à janela de Movimentos do Módulo de Contabilidade, directamente a partir do Editor de Vendas, pertencente ao módulo de Vendas. Esta possibilidade existe não só entre os vários módulos do ERP, mas também em aplicações externas que podem utilizar os diferentes Motores da Aplicação.



Figura 23: ERP Primavera - Arquitectura de 3 Níveis

Arquitectura dos Motores

A arquitectura do motor do ERP Primavera, estruturada em três camadas, isola o acesso aos dados, potenciando a sua reutilização por aplicações externas, sem colocar em causa a integridade dos dados. (Primavera BSS, 2007)

O motor de uma aplicação é composto por um conjunto de DLLs⁶ (*Dynamic Link Library*) COM que podem ser referenciadas e utilizadas a partir de aplicações externas, bastando para tal que estejam registadas no sistema e que sejam incluídas no respectivo projecto.

A Figura 24 apresenta a arquitectura e organização do motor de um ou mais módulos:

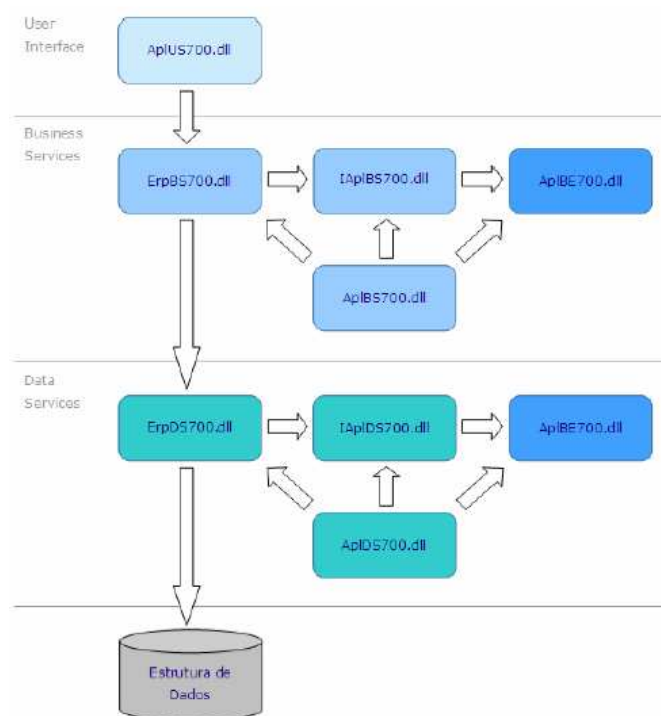


Figura 24: ERP Primavera - Arquitectura do Motor

A arquitectura do motor é composta pelas seguintes DLLs:

- ✓ **AplBE700.dll** – Objectos de Negócio: representam as entidades geridas pelo motor e pela aplicação;
- ✓ **ErpBS700.dll** – permite o acesso aos serviços de negócio de cada Módulo;
- ✓ **IAplBS700.dll** – Implementam os interfaces para os serviços de negócio;

⁶ Implementação feita pela Microsoft para o conceito de bibliotecas compartilhadas nos sistemas operativos Microsoft Windows e OS/2.

- ✓ **AplBS700.dll** – Serviços de Negócio: Disponibilizam todas as regras de negócio;
- ✓ **IAplIDS700.dll** – Implementam os interfaces para os serviços de dados;
- ✓ **ApIDS700.dll** – Serviços de Dados: disponibilizam todos os serviços de carregamento e gravação de dados a partir da base de dados.

A sigla Apl (Aplicação) deverá ser substituída pela nomenclatura associada a cada módulo ou área. Por exemplo, para o módulo de Contabilidade deverá ser ICbIBS700, CbIBE700, etc.

Objectos ou Entidades de Negócio (BE)

Os objectos/entidades de negócio geridos por cada aplicação, são disponibilizados através de objectos simples com um conjunto de propriedades na DLL AplIBE700.dll.

A nomenclatura dos objectos é a seguinte:

<Apl>BE<Nome>

Como exemplos, pode-se destacar os seguintes casos:

- ✓ GcpBEDocumentoVenda (área de Logística & Tesouraria)
- ✓ GcpBEDocumentoCompra (área de Logística & Tesouraria)
- ✓ CbIBEConta (área de Contabilidade)

Ainda a título de exemplo, na figura abaixo, estão listadas as propriedades associadas ao objecto *GcpBEDocumentoVenda*, compostas por um conjunto de propriedades que definem o cabeçalho do documento e pelas linhas do documento:



Figura 25: ERP Primavera - Propriedades do objecto GcpBEDocumentoVenda

Serviços de Negócio (BS)

A DLL ERPBS700.dll é o ponto de entrada no motor do ERP e disponibiliza o acesso a todos os serviços do motor para cada módulo que implementa todas as regras de negócio.

Serviços de Dados (DS)

A DLL IApIDS700.dll disponibiliza os serviços de interacção com a base de dados.

Os serviços deste componente têm uma correspondência directa com os mesmos serviços dos Serviços de Negócio (BS). Sempre que um serviço do BS necessita de aceder à base de dados, fá-lo através do serviço correspondente no DS.

3.1.2 Visual Basic for Applications (VBA)

O *Visual Basic for Applications* (VBA) é uma implementação do *Visual Basic* da *Microsoft*, que se encontra incorporada em todos os programas do *Microsoft Office*. Além disso, existem outros programas que se apropriaram da linguagem, devido à sua flexibilidade.

Esta tecnologia, largamente divulgada como uma ferramenta de programação, é utilizada pelas soluções Primavera, que a disponibilizam como tecnologia principal de extensibilidade.

Através do VBA, é possível estender as soluções Primavera, mediante a criação de funcionalidades específicas, que não vêm de base com o ERP, nomeadamente:

- ✓ Aceder aos próprios objectos mantidos e manipulados no contexto de cada uma das principais operações – manutenção de clientes, editores de vendas, etc.;
- ✓ Associar código a determinados eventos despoletados pelas operações mais importantes das aplicações;
- ✓ Criar separadores do utilizador para edição dos campos do utilizador associados a cada entidade;
- ✓ Criar formulários do utilizador;
- ✓ Integrar os desenvolvimentos realizados na solução Primavera, via Funções do Utilizador que podem ser associadas ao Navegador ou a Menus do Utilizador.

Para trabalhar com o VBA, é utilizada uma interface denominada *Integrated Development Environment* (IDE). Todas as ferramentas de desenvolvimento do VBA podem ser acedidas a partir

deste IDE, incluindo o *Project Explorer*, o *Object Browser*, o editor de código, a interface de desenho de formulários e o *debugger*.

Os projectos VBA, além de conterem os itens do projecto, armazenam todo o código VBA.

Um projecto VBA pode conter:

- ✓ Módulos de Código (*Code Modules*): contêm declarações, procedimentos e funções;
- ✓ Classes (*Class Modules*): permitem programação orientada a objectos, agrupando procedimentos, funções e declarações numa classe;
- ✓ Formulários (*Forms*): permitem a criação de interfaces poderosos;
- ✓ Referências (*References*): permitem a reutilização de código a partir de outros projectos.

3.2 OpenERP

3.2.1 Caracterização da solução

O OpenERP – inicialmente conhecido por Tiny ERP – é considerado actualmente um dos sistemas ERP *Open Source* com a maior taxa de crescimento no mundo e destaca-se pela sua arquitectura elegante, pelo uso de ferramentas ágeis de desenvolvimento, pela sua elevada flexibilidade e pelo custo reduzido.

Como a maioria das aplicações de *software* livre de larga escala, a acessibilidade, a flexibilidade e a simplicidade são os principais pilares do seu desenvolvimento, *deployment* e utilização.

Segundo Serrano e Sarriegi (2006), citados por Fontana *et al.* (2008, p.2), a utilização de sistemas ERP *open source* está-se a tornar uma alternativa viável para micros e pequenas empresas, visto que esses sistemas possuem funcionalidades já implementadas que podem atender às suas necessidades de negócio e apresentam facilidades como adaptabilidade crescente, dependência decrescente de um único fornecedor e custos reduzidos. Entretanto, tanto os ERP proprietários como os *open source* envolvem implementações complexas que geralmente requerem mudanças nos processos organizacionais e/ou adaptações das funcionalidades do ERP.

Uma das grandes vantagens do *software* livre, é o facto de permitir reduzir significativamente os custos do desenvolvimento, através da reutilização de bibliotecas e códigos

fonte livres. A responsabilidade do desenvolvimento dos sistemas está a cargo de comunidades de programadores e analistas, que partilham a troca de conhecimentos e soluções tecnológicas.

Segundo a *Free Software Foundation*, existem quatro tipos de liberdade que os utilizadores do *software* livre têm em relação ao *software* proprietário:

- ✓ A liberdade de executar o programa para qualquer propósito;
- ✓ A liberdade de estudar como o programa funciona e adaptá-lo às suas necessidades. O acesso ao código fonte é uma pré-condição para que isto aconteça;
- ✓ A liberdade de distribuir cópias, para ajudar outras pessoas;
- ✓ A liberdade de introduzir melhorias no programa, e disponibilizar essas melhorias para o público, para que toda a comunidade seja beneficiada.

As seguintes desvantagens são apontadas às soluções proprietárias:

- ✓ Custos de manutenção da solução;
- ✓ Limitação do número de utilizadores;
- ✓ Custos com formação;
- ✓ Custos com a actualização das versões;
- ✓ Necessidade de personalização de alguns procedimentos;
- ✓ Custos de criação de módulos específicos.

Estima-se que o custo com as soluções proprietárias é entre 20 a 50% mais elevado que as soluções *Open Source*, mas o ponto essencial reside na falta de flexibilidade ligada ao código fonte protegido, o que implica uma maior rigidez das empresas que utilizam essas soluções.

Existem vários sistemas ERP *open source* disponíveis no mercado, entre os quais:

- ✓ Compiere;
- ✓ ERP 5;
- ✓ JFire;
- ✓ OpenERP;
- ✓ Openbravo;
- ✓ WebERP.

O OpenERP é uma *Framework* com dezenas de módulos desenvolvidos por especialistas de diferentes áreas que permitem uma personalização para a realidade de cada entidade que a vai utilizar. (Pinckaers e Gardiner, 2009)

Os módulos mais utilizados são: Contabilidade, Gestão de Stocks, Gestão de Vendas e Compras, Gestão de Projectos, Gestão de Recursos Humanos, Marketing, Ponto de Vendas e CRM.

O OpenERP é considerado o único ERP utilizado pelos grandes grupos empresariais e pelas PME, oferecendo uma grande diversidade de serviços tais como Hotéis, Restaurantes, Informática, Agro-alimentar, Têxteis, Produção, Farmacêutico, etc.

O produto é destacado pela sua qualidade, pela opção tecnológica e pela representação a nível mundial, que vai da China à Europa, e que recentemente chegou a Cabo Verde.

O OpenERP é um *software* com licença GPL⁷ (*General Public License*), o que permite a qualquer pessoa a utilização, duplicação e divulgação (oferecer ou vender) desse *software*, sem restrições.



Figura 26: Ambiente de Trabalho do OpenERP

Os custos de aquisição do *software* são considerados reduzidos, pois não contemplam encargos comerciais, nem de marketing. A construção do *software* é mais barata porque beneficia de códigos disponíveis e da contribuição de milhares de programadores.

⁷ GPL é a licença para programas da *Free Software Foundation*

Segundo informações disponíveis no site oficial do OpenERP (<http://www.openerp.com>), o *software* está actualmente disponível em 18 idiomas e possui uma rede mundial com mais de 800 colaboradores.

Em suma, as seguintes vantagens são apontadas ao OpenERP:

- ✓ 100% *software* livre;
- ✓ Comunidade de desenvolvimento activa (documentação *on-line* disponível);
- ✓ Independência entre módulos;
- ✓ Internacionalização completa;
- ✓ Permite personalizações sem recorrer à programação;
- ✓ Disponibiliza cerca de 350 módulos.

3.2.2 Arquitectura do OpenERP

Pinckaers e Gardiner (2009), descrevem o OpenERP como um sistema cliente - servidor que pode ser acedido a partir de um *browser*, ou através de um aplicativo cliente instalado em cada posto de trabalho. Os dois métodos de acesso são semelhantes e ambos podem ser utilizados, simultaneamente, a partir do mesmo servidor. O acesso via *browser* é mais recomendado caso o servidor OpenERP esteja fisicamente num local remoto, pois este apresenta uma melhor tolerância em relação aos tempos de espera. Além disso, o *browser* vem incorporado no sistema operativo, o que elimina o trabalho de instalação e manutenção dos postos de trabalho.

Por outro lado, a aplicação cliente do OpenERP (conhecida como cliente GTK, por causa da tecnologia com que é construída) apresenta um melhor desempenho quando o servidor estiver localizado na rede local.

Segundo Pinckaers e Gardiner (2009), a arquitectura do OpenERP é composta por 3 componentes principais:

1. Servidor de base de dados *PostgreSQL*, que contém todas as bases de dados;
2. Servidor de aplicação OpenERP, escrito na linguagem de programação *Python*⁸, que contém toda a lógica do negócio;
3. Servidor de Internet, que permite o acesso via *browser*.

⁸ *Python* é uma linguagem de programação de alto nível, interpretada, imperativa, extensível e orientada a objectos.

A figura abaixo, resume a arquitectura do OpenERP:

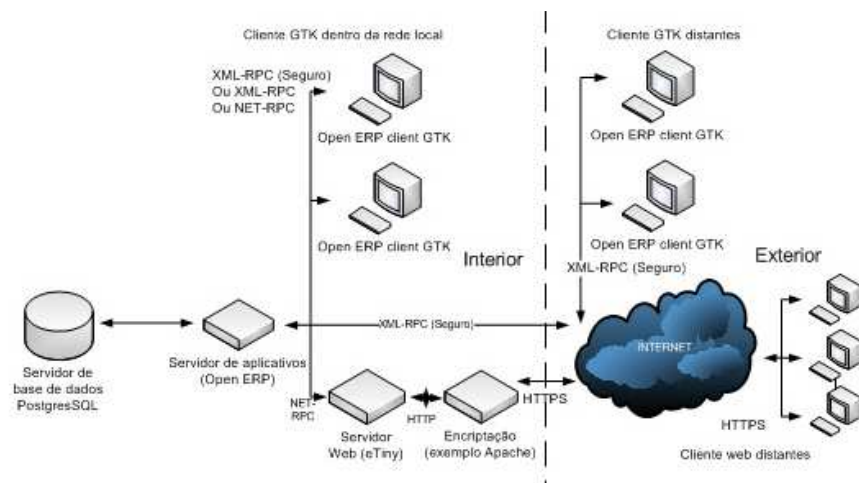


Figura 27: Arquitectura do OpenERP (Fonte: <http://www.openerp.com/services/books>)

O sistema é compatível com os sistemas operativos *Windows* e *Linux*, existindo um pacote de instalação para cada um dos casos.

4 Arquitectura para a integração OpenERP – Primavera

Neste capítulo irá ser apresentada a arquitectura do sistema de integração, as suas funcionalidades e ferramentas tecnológicas adoptadas, bem como os resultados obtidos através da realização de alguns testes.

4.1 Arquitectura da solução

A interface desenvolvida tem por objectivo integrar os sistemas OpenERP e Primavera, no que respeita a fichas de Clientes e Artigos, e movimentos financeiros relativos a Documentos de Venda (Facturas e Notas de Crédito) e Recebimentos de Clientes.

Os movimentos têm origem no sistema OpenERP, existindo um automatismo que procederá à geração de ficheiros XML com a informação desejada, para posterior descarga no sistema Primavera (Figura 28).

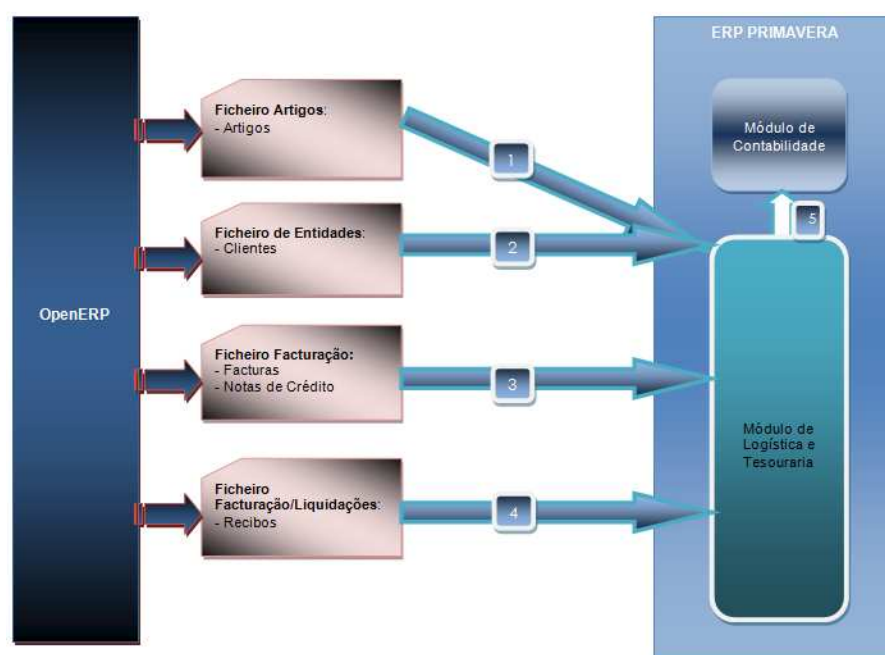


Figura 28: Arquitectura da solução

De acordo com a figura acima, a integração será baseada na geração de 5 tipos de ficheiros pelo OpenERP, nomeadamente:

1. Um ficheiro com os dados dos Artigos criados ou actualizados no OpenERP, desde a última integração. A integração deste ficheiro resultará na criação ou actualização da ficha do Artigo no sistema Primavera;
2. Um ficheiro com os dados das entidades (Clientes) criadas ou actualizadas no OpenERP. A integração deste ficheiro resultará na criação ou actualização da entidade no sistema Primavera;
3. Um Documento de Venda na Conta Corrente do Cliente em questão, representando a Factura (ou Nota de Crédito) do cliente, detalhada com as diversas linhas de artigos. Este documento será posteriormente descarregado na Contabilidade (processo 5), através de mecanismos disponibilizados no próprio ERP Primavera;
4. Um ficheiro com as liquidações dos documentos integrados pelo ficheiro descrito no processo anterior. A integração do ficheiro originará um Recibo ou Pagamento na Conta Corrente do Cliente em questão, sendo a descarga na Contabilidade efectuada posteriormente, pelo mesmo processo descrito anteriormente (processo 5).

Para uma melhor contextualização do processo de integração, de seguida serão apresentados os relacionamentos entre as principais tabelas da base de dados do ERP Primavera, onde os ficheiros serão integrados. Os esquemas apresentados nas Figuras 29 e 30 correspondem, respectivamente, ao cenário das vendas e dos recebimentos, que estão incluídos no módulo de Logística e Tesouraria.

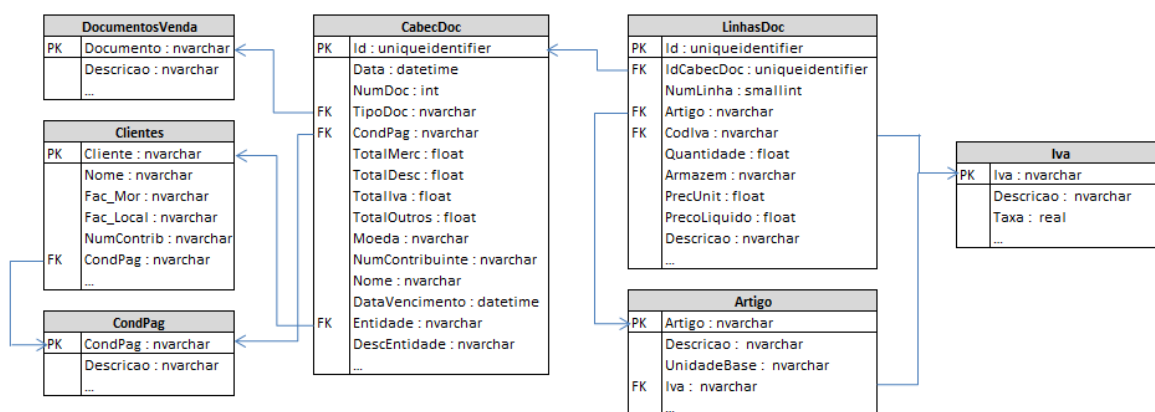


Figura 29: Diagrama Relacional das Vendas no ERP Primavera

Um documento de venda é constituído por um cabeçalho (CabecDoc) e por uma ou várias linhas (LinhasDoc). A tabela CabecDoc contém o registo de todos os cabeçalhos dos documentos de

venda emitidos, incluindo campos como a data de emissão do documento, tipo e número do documento, condições de pagamento acordadas com o cliente, etc.

Por sua vez, a tabela LinhasDoc contém o registo referente às linhas relacionadas com os vários itens (Artigos) que constam de um documento de venda.

Estas duas tabelas relacionam-se entre si, na medida em que as linhas existentes na tabela LinhasDoc correspondem a um cabeçalho existente na tabela CabecDoc.

Além destas tabelas, é preciso ter em conta as seguintes: Clientes; Artigo; DocumentosVenda (tabela de registo dos tipos de documentos de venda e respectivas configurações); CondPag (registo das possíveis condições de pagamento que poderão ser negociadas com os clientes) e Iva.

No cenário dos recebimentos de clientes (Figura 30), a tabela CabLiq contém o registo dos cabeçalhos das liquidações efectuadas, enquanto que a tabela LinhasLiq apresenta o registo das linhas das liquidações, ou seja, informações sobre os documentos que foram liquidados, os respectivos valores, etc. Relativamente às restantes tabelas, DocumentosCCT é a tabela de registo dos tipos de documento de contas correntes e respectivas configurações; Pendentes é onde são registados os documentos pendentes para liquidação; a tabela Historico contém o registo de todos os movimentos em conta corrente; MovimentosBancos contém os movimentos bancários utilizados no acto da liquidação e está interligada com a tabela ContasBancarias que, por sua vez, está associada à tabela Bancos.

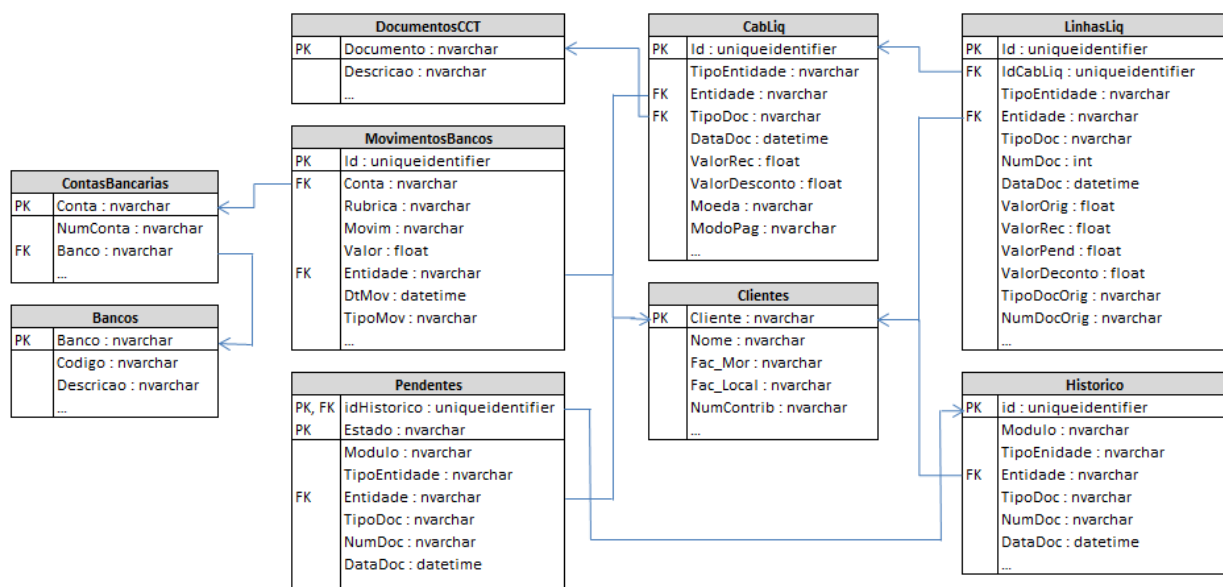


Figura 30: Diagrama Relacional dos Recebimentos no ERP Primavera

Conforme foi referido anteriormente, o propósito deste trabalho é integrar os módulos de Vendas e Recebimentos do OpenERP, com o módulo de Contabilidade do ERP Primavera. O processo utilizado consistiu em exportar os movimentos do OpenERP para as Vendas e Recebimentos do ERP Primavera, sendo que a última etapa traduz-se na descarga dos documentos no módulo de Contabilidade do ERP Primavera, através de mecanismos disponibilizados pelo próprio ERP (Figura 31). Nesta fase, os registos são armazenados nas tabelas CabecMovCBL (cada registo desta tabela define o cabeçalho de um documento contabilístico) e Movimentos (permite o registo do lançamento dos diversos movimentos que compõem um documento). Previamente, as contas deverão estar criadas no Plano de Contas (tabela PlanoContas).

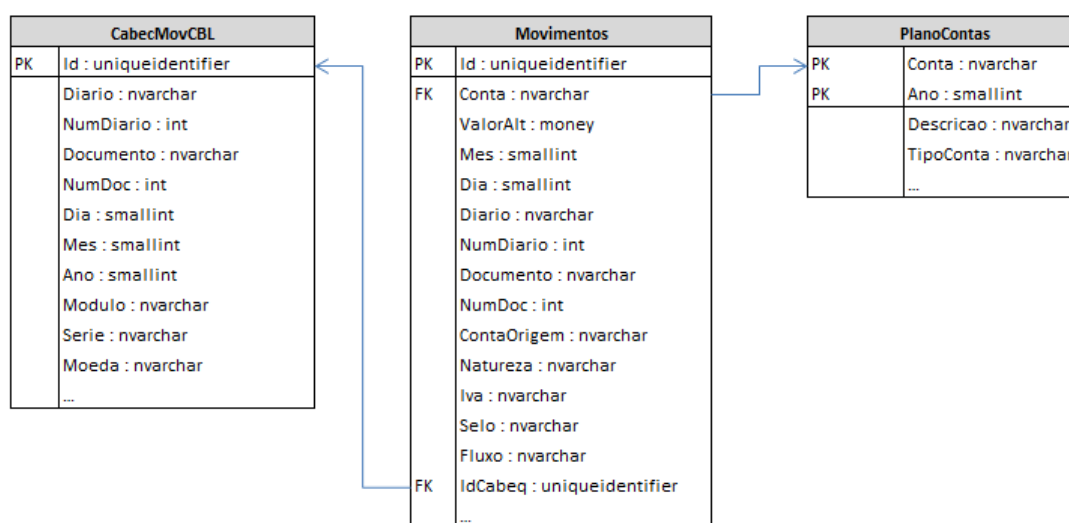


Figura 31: Tabelas movimentadas na integração com o módulo de Contabilidade do ERP Primavera

4.2 Descrição do Processo de Integração

Depois de os documentos terem sido gerados e exportados pelo OpenERP, o processo de integração no ERP Primavera consistirá nos seguintes passos:

- O utilizador dará a ordem de leitura dos ficheiros previamente colocados numa directoria específica, a parametrizar para o efeito;
- A Interface Primavera fará, de seguida, o *parsing* dos ficheiros, de acordo com regras específicas assentes na estrutura dos mesmos, definida na secção seguinte, e irá gerar os movimentos nos módulos Primavera, com base no *parsing* efectuado;
- Cada ficheiro processado com sucesso, será movido para uma nova directoria a parametrizar – Directoria de Ficheiros Processados;

- d) Quaisquer erros que ocorram no processamento serão registados num ficheiro de *Log*, com a informação necessária à correcta identificação dos mesmos.

O processo será disponibilizado através de uma interface consistente com a interface da solução Primavera v7.55, a disponibilizar através de menu próprio (Menu de Utilizador), no módulo de Logística e Tesouraria.

As Figuras 32 e 33 ilustram o processo de integração acima descrito:

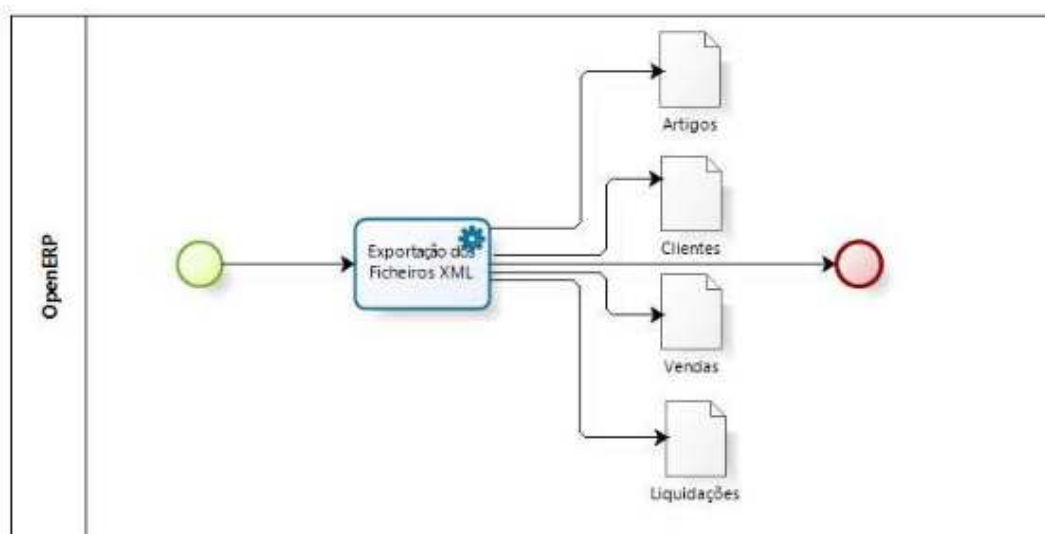


Figura 32: Ficheiros gerados pelo OpenERP

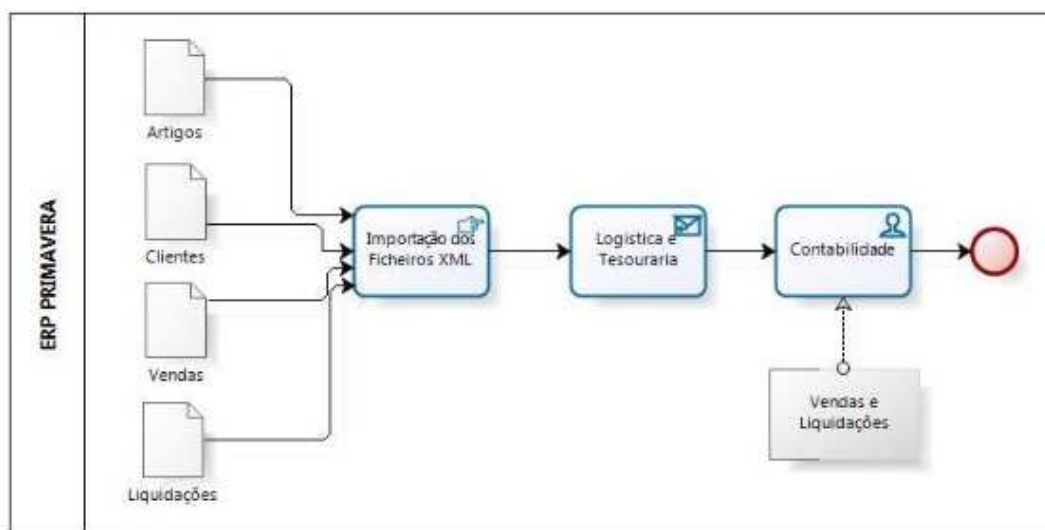


Figura 33: Importação dos ficheiros no Primavera e integração na Contabilidade

4.3 Especificação da estrutura dos ficheiros XML

A integração no sistema Primavera só será exequível a partir do momento em que o sistema OpenERP esteja devidamente parametrizado e configurado para gerar os ficheiros necessários.

Os atributos do tipo Data e Número, devem respeitar o seguinte formato:

- ✓ Datas – AAAA-MM-DD
- ✓ Numéricos (Valores e Percentagens) – 0.00

4.3.1 Ficheiro de Artigos

Estes ficheiros incluem informação acerca dos artigos que foram criados ou actualizados no último período no sistema OpenERP, resultando a sua integração na criação ou actualização da respectiva ficha do artigo no sistema Primavera. O principal objectivo é garantir a uniformidade dos códigos nos dois sistemas.

Nomenclatura: **ARTAAAAMMDD.xml** (ex: ART20100623.xml)

No caso deste ficheiro, são apenas dois os campos enviados para o Primavera, sendo os restantes preenchidos pela utilização de um *artigo template*, uma vez que a informação constante dos dois sistemas é substancialmente distinta.

A unidade base dos artigos não é enviada para o Primavera. Assumiu-se que todos os artigos ficarão associados à unidade “UN”.

CodArtigo	Representa o Código da Entidade no sistema Primavera
Descricao	Nome do Artigo
TaxaIVA	Taxa de IVA a aplicar ao Artigo

Tabela 1: Campos da entidade Artigo

A Figura 34 representa o *XML Schema* para a entidade Artigo:

```
<?xml version="1.0" encoding="utf-8" ?>
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Artigos">
    <xs:complexType>
      <xs:sequence>
        <xs:element maxOccurs="unbounded" name="Artigo">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="CodArtigo" type="xs:string" />
              <xs:element name="Descricao" type="xs:string" />
              <xs:element name="TaxaIVA" type="xs:decimal" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Figura 34: *XML Schema* da entidade Artigo

4.3.2 Ficheiro de Clientes

Estes ficheiros incluem informação acerca dos clientes que foram criados ou actualizados no período em questão, no sistema OpenERP, e resultará na criação ou actualização da respectiva ficha no sistema Primavera. Têm como principal objectivo garantir a uniformidade dos códigos nos dois sistemas.

Nomenclatura: **CLIAAAAMMDD.xml** (Ex: CLI20100622.xml)

A tabela abaixo descremina os campos que serão enviados para o sistema Primavera, pelo OpenERP. No entanto, no sistema Primavera existem um conjunto de campos adicionais que terão que ser actualizados manualmente pelo utilizador, uma vez que se tratam de campos muito específicos deste sistema. Na criação do cliente, será também usado um *cliente template*, que servirá de base à criação do novo cliente, sendo a maior parte dos atributos preenchidos com base nesse *template*, podendo posteriormente o utilizador proceder aos ajustes necessários.

CodEntidade	Representa o Código da Entidade no sistema Primavera
Nome	Nome do Cliente
Fac_Mor	Morada
Fac_Local	Localidade
Fac_Cp	Código Postal
Fac_Cploc	Localidade do Código Postal
Fac_Tel	Telefone
Fac_Fax	Fax
NumContrib	Nº de Contribuinte

Tabela 2: Campos da entidade Cliente

A Figura 35 representa o *XML Schema* para a entidade Cliente:

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Entidades">
    <xs:complexType>
      <xs:sequence>
        <xs:element maxOccurs="unbounded" name="Entidade">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="CodEntidade" type="xs:string" />
              <xs:element name="Nome" type="xs:string" />
              <xs:element name="Fac_Mor" type="xs:string" />
              <xs:element name="Fac_Local" type="xs:string" />
              <xs:element name="Fac_CP" type="xs:unsignedShort" />
              <xs:element name="Fac_CPLoc" type="xs:string" />
              <xs:element name="Fac_Tel" type="xs:unsignedInt" />
              <xs:element name="Fac_Fax" type="xs:unsignedInt" />
              <xs:element name="NumContrib" type="xs:unsignedInt" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Figura 35: *XML Schema* da entidade Cliente

4.3.3 Ficheiro de Facturação

Este ficheiro enviará para o sistema Primavera, informação sobre os documentos de venda emitidos para clientes, e permitirá dessa forma e se for pretendido, gerir a sua conta corrente a partir do sistema Primavera, para além da sua gestão Contabilística. Os documentos devem ser lançados na moeda base CVE (Escudos Cabo-Verdianos).

Nomenclatura: **VENAAAAMMDD.xml** (Ex: VEN20100622.xml)

Os ficheiros deste tipo, incluem todas as Facturas e Notas de Crédito efectuadas no período em questão, quer tenham sido liquidadas ou não (no sistema OpenERP), motivo pelo qual será também necessário emitir um ficheiro que identifique as liquidações efectuadas.

Cada documento gerado no sistema Primavera (Factura ou Nota de Crédito), representará uma cópia do documento original lançado no OpenERP, com o mesmo número de linhas e artigos, discriminando a taxa de IVA aplicada.

A informação proveniente do ficheiro será a seguinte:

CodCliente	Deverá identificar o código do Cliente do módulo de Logística no sistema Primavera
TipoDocumento	Deverá conter um dos seguintes valores: - F: Fatura; - C: Nota de Crédito;
NumDocOpenERP	Nº do documento no sistema OpenERP
DataDoc	AAAA-MM-DD. Data de emissão do Documento.
DataVenc	AAAA-MM-DD. Data de Vencimento do Documento.
DescCliente	Percentagem de Desconto de Cliente
CodArtigo	Código do Artigo no sistema OpenERP
PrecUnit	Preço Unitário do Artigo
Quantidade	Quantidade na Unidade Base do Artigo
DescArtigo	Percentagem de desconto associado ao artigo

Tabela 3: Campos do ficheiro de Facturação

A Figura 36 apresenta o *XML Schema* dos documentos de venda:

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Vendas">
    <xs:complexType>
      <xs:sequence>
        <xs:element maxOccurs="unbounded" name="Documento">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Cabecalho">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="CodCliente" type="xs:string" />
                    <xs:element name="TipoDocumento" type="xs:string" />
                    <xs:element name="NumDocOpenERP" type="xs:string" />
                    <xs:element name="DataDoc" type="xs:date" />
                    <xs:element name="DataVenc" type="xs:date" />
                    <xs:element name="DescCliente" type="xs:decimal" />
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
              <xs:element name="Linhas">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="Linha">
                      <xs:complexType>
                        <xs:sequence>
                          <xs:element name="CodArtigo" type="xs:string" />
                          <xs:element name="PrecUnit" type="xs:decimal" />
                          <xs:element name="Quantidade" type="xs:decimal" />
                          <xs:element name="DescArtigo" type="xs:decimal" />
                        </xs:sequence>
                      </xs:complexType>
                    </xs:element>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Figura 36: XML Schema dos documentos de venda

4.3.4 Ficheiro de Liquidações

Este ficheiro enviará para o sistema Primavera, informação sobre as liquidações dos documentos de venda efectuadas no sistema OpenERP. A Interface Primavera irá gerar um recibo (ou pagamento, caso o valor liquidado seja menor do que 0) por cada liquidação efectuada no OpenERP.

Nomenclatura: **LIQ**AAAAMMDD.xml (Ex: LIQ20100622.xml)

O ficheiro deverá obedecer aos seguintes critérios:

- ✓ Podendo o ficheiro apresentar várias liquidações, é necessário que este se apresente ordenado pela data de liquidação;
- ✓ É necessário que os clientes estejam parametrizados para utilizar a moeda base do sistema (CVE).

A informação disponibilizada no ficheiro está representada na Tabela 4 e o *XML Schema* está esquematizada na Figura 37:

CodCliente	Código do cliente no OpenERP
DataLiq	Data da Liquidação
ValorLiq	Valor da Liquidação: Positivo caso represente um valor a receber do cliente ou negativo caso represente um valor a pagar ao cliente
NumDocOpenERP	Representa o nº da Factura/Nota de Crédito no sistema OpenERP, a liquidar.
TipoDocumento	Identifica o tipo de documento. Deverá conter um dos seguintes valores: - F : Factura; - C : Nota de Crédito;
ValorLiqDoc	Valor (da Factura ou Nota de Crédito) a liquidar
ModoRecPag	Código do Modo de Recebimento ou Pagamento no sistema Primavera, parametrizado no sistema OpenERP, em campo específico;
ValorRecPag	Valor do Recebimento/Pagamento associado ao Modo de Recebimento/Pagamento
NumCheque	Número do Cheque, caso o recebimento/pagamento seja efectuado por cheque

Tabela 4: Campos do ficheiro de liquidações

```

<?xml version="1.0" encoding="Windows-1252"?>
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Liquidacoes">
    <xs:complexType>
      <xs:sequence>
        <xs:element maxOccurs="unbounded" name="Documento">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Cabecalho">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="CodCliente" type="xs:string" />
                    <xs:element name="DataLiq" type="xs:date" />
                    <xs:element name="ValorLiq" type="xs:decimal" />
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
              <xs:element name="Linhas">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="Linha">
                      <xs:complexType>
                        <xs:sequence>
                          <xs:element name="NumDocOpenERP" type="xs:unsignedByte" />
                          <xs:element name="TipoDocumento" type="xs:string" />
                          <xs:element name="ValorLiqDoc" type="xs:decimal" />
                        </xs:sequence>
                      </xs:complexType>
                    </xs:element>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
              <xs:element name="ModosRecPag">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="ModoRecPag">
                      <xs:complexType>
                        <xs:sequence>
                          <xs:element name="ModoRecPag" type="xs:string" />
                          <xs:element name="ValorRecPag" type="xs:decimal" />
                          <xs:element name="NumCheque" type="xs:decimal" />
                        </xs:sequence>
                      </xs:complexType>
                    </xs:element>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>

```

Figura 37: XML Schema dos documentos de liquidação

4.4 Ferramentas Utilizadas

No seguimento do que foi referido nas secções de apresentação das características e arquitecturas do ERP Primavera (Secção 3.1) e do OpenERP (Secção 3.2), as ferramentas utilizadas por esses dois sistemas, foram as escolhidas para implementar a arquitectura de integração.

Do lado do OpenERP foram desenvolvidos scripts utilizando as linguagens Python e XML, que permitiram gerar os ficheiros finais, em formato XML, com a estrutura pretendida. Foi utilizada a versão 5.0.15 do OpenERP, para Windows.

Do lado do ERP Primavera, o projecto foi desenvolvido com base na Tecnologia de Integração Primavera (TIP), que permite aceder aos motores utilizados pelo próprio ERP, garantindo assim a fiabilidade e consistência dos dados. O ambiente de desenvolvimento utilizado foi o Microsoft Visual Studio 6 (Microsoft Visual Basic 6), por se tratar da ferramenta de desenvolvimento utilizada pela Primavera e por incluir um conjunto de funcionalidades que facilitam a integração de componentes COM. Também foi utilizado o VBA e o Microsoft SQL Server 2008. A versão do ERP utilizada foi a Executive v7.55. No entanto, os componentes dos motores Primavera podem ser utilizados em outros ambientes de programação.

Além das ferramentas acima mencionadas, foram utilizadas algumas funcionalidades do ERP Primavera que facilitam a extensibilidade da solução standard, que serão a seguir discriminadas.

4.4.1 ERP Primavera: Campos e Tabelas do Utilizador

A estrutura da base de dados do ERP Primavera pode ser adaptada através da criação de novos campos de utilizador (CDU), em tabelas do sistema, ou criando novas tabelas de utilizador (TDU) para armazenar dados adicionais.

Assistente de Criação de Tabelas e Campos do Utilizador

Dados do Campo do Utilizador
Indique a tabela e os dados do campo do utilizador.

Indique a tabela onde será inserido o novo campo do utilizador.

Tabela: Clientes

Campo: CDU_Auxiliar

Tipo: NVARCHAR Dimensão: 10

Descrição: Auxiliar

Título: Auxiliar

Valor defeito:

☒ Visível ☐ Criar índice sobre este campo
☒ Pode ter valores nulos ☐ Chave primária

Anterior Próximo > Cancelar

Figura 38: Assistente de criação de Tabelas e Campos do Utilizador

Os campos de utilizador têm a nomenclatura CDU_NomeDoCampo, enquanto que a nomenclatura das tabelas é TDU_NomeDaTabela.

4.4.2 ERP Primavera: Funções do Utilizador

De acordo com a Primavera BSS (2007), uma função de utilizador corresponde à definição de uma acção (envio de mensagens de email ou SMS personalizados para cada empresa, execução de Macros VBA, etc.) que pode interagir com o utilizador e com o sistema para automatizar uma determinada tarefa. Este conceito permite estender a solução Primavera para garantir todas as regras de negócio específicas a cada empresa.

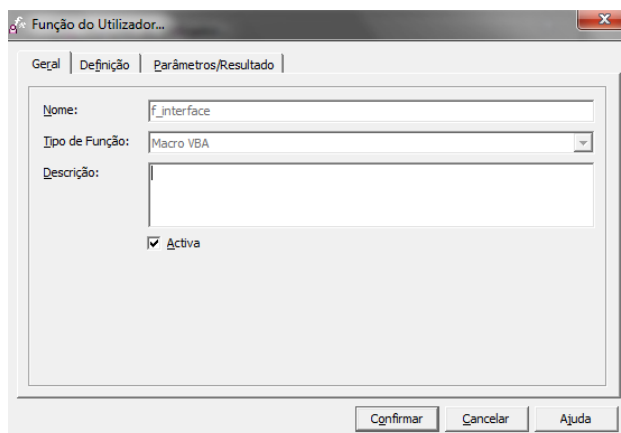


Figura 39: Assistente de criação de uma Função do Utilizador

4.4.3 ERP Primavera: Menus do Utilizador

Os Menus do Utilizador permitem complementar os menus das aplicações para integrarem novas opções que executarão uma Função de Utilizador que, por exemplo, pode abrir uma aplicação externa ou iniciar uma operação em *background*.

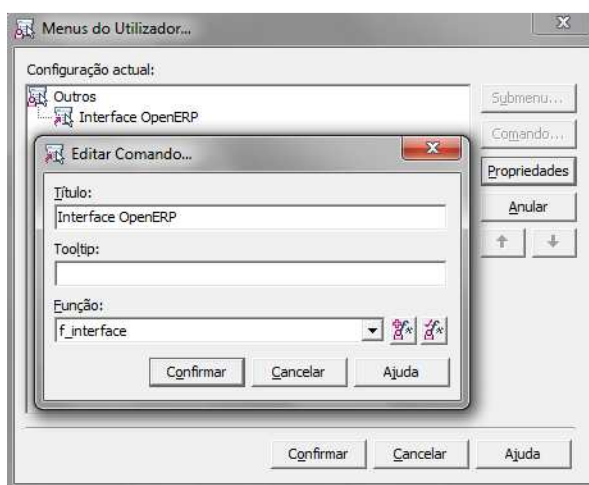


Figura 40: Criação de Menus de Utilizador

4.5 Estrutura das Pastas da Interface

A figura abaixo apresenta a estrutura das pastas que constituem a Interface:

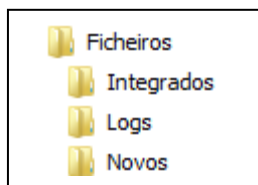


Figura 41: Estrutura das pastas

Para o correcto funcionamento do sistema, os nomes das pastas *Integrados*, *Logs* e *Novos* não devem ser alterados.

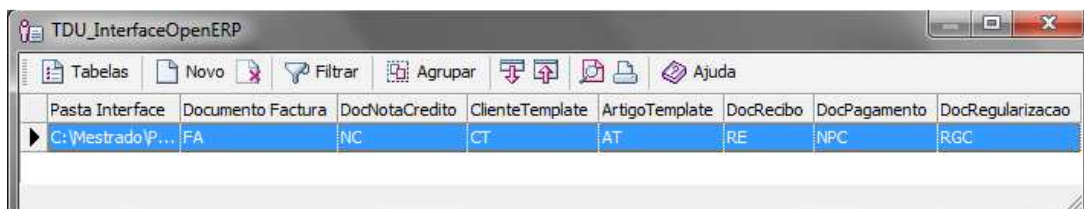
4.6 Estrutura da Tabela de Parametrização

Para a especificação da tabela com as parametrizações necessárias, recorreu-se à funcionalidade de criação de Tabelas e Campos de Utilizador, apresentada na secção 4.4.1. Conforme foi referido, a grande vantagem desta funcionalidade é o facto de permitir manter dados não suportados pela solução *standard*.

A tabela foi denominada de **TDU_InterfaceOpenERP** e contém os seguintes campos:

- ✓ **CDU_PastaInterface:** deve conter o caminho da localização dos ficheiros no disco;
- ✓ **CDU_DocFactura:** deve conter o código do documento Factura, a gerar no ERP Primavera;
- ✓ **CDU_DocNotaCredito:** código do documento Nota de Crédito, a gerar no ERP Primavera;
- ✓ **CDU_ClienteTemplate:** código do *cliente template*, que servirá de base à criação do novo cliente, sendo a maior parte dos atributos preenchidos com base nesse *template*;
- ✓ **CDU_ArtigoTemplate:** código do *artigo template*, que servirá de base à criação do novo artigo;
- ✓ **CDU_DocRecibo:** código do documento de Recebimento de Clientes, a gerar no ERP Primavera;
- ✓ **CDU_DocPagamento:** código do documento de Pagamento a Clientes, a gerar no ERP Primavera;

- ✓ **CDU_DocRegularizacao:** código do documento de Regularização de Clientes, a gerar no ERP Primavera. Este documento só é utilizado quando o valor a débito é igual ao valor a crédito, ou seja, o saldo é zero.



Pasta Interface	Documento Factura	DocNotaCredito	ClienteTemplate	ArtigoTemplate	DocRecibo	DocPagamento	DocRegularizacao
C:\Mestrado P...	FA	NC	CT	AT	RE	NPC	RGC

Figura 42: Tabela de Utilizador TDU_InterfaceOpenERP

4.7 Resultados obtidos

Nesta secção será apresentado um processo completo de integração, desde a geração dos ficheiros no OpenERP, até à integração no ERP Primavera. Para tal, será considerado, como caso de estudo, a facturação de uma empresa fictícia, dedicada à comercialização de produtos informáticos.

Normalmente, a implementação do módulo de vendas de um ERP numa organização, começa com a criação dos artigos a serem comercializados, e dos clientes.

4.7.1 Operações a serem realizadas no OpenERP

Inicialmente serão criados 2 artigos, com informação básica. Os códigos e as descrições serão A1 - Rato Wireless e A2 - Mala p/ Portátil (Figura 43). Além disso, será especificada uma Taxa de Iva de 15%.



Descrição do artigo	Códigos
Nome: Rato Wireless	Código: A1

Descrição do artigo	Códigos
Nome: Mala p/ Portátil	Código: A2

Figura 43: Artigos criados no OpenERP

Será também criado um cliente com o código C1, de nome António Silva, residente na Rua Cidade de Lisboa, Cidade da Praia. Nesta ficha de cliente, serão também preenchidos os campos referentes aos Números de Telefone, Fax, Número de Contribuinte e Código Postal (Figura 44):

OpenERP
Made by Tiny & Axelor

MAIN MENU **SHORTCUTS** Modules

Terceiros Search Form Calendar Gantt Gra

New Edit Duplicate Delete << First < Previous

Nome: António Silva Código: C1 ? Cliente: ☒
 Título: LDA ? Linguagem: Portugese / português ? Fornecedor: ☐

Geral Vendas & Compras Histórico Notas Contabilidade Nif

Partner Contacts

Nome do contacto: António Silva Type:
 Função: ? Tipo de endereço:
 Rua: Rua Cidade de Lisboa Rua2:
 Código postal: 417 Cidade: Praia
 País: Fed. State:
 Telefone: 9977345 Fax: 9977346
 Telemóvel: E-Mail:

Figura 44: Confirmação da criação do cliente C1 (António Silva)

O próximo passo consistirá na emissão de um documento de venda (Factura a Crédito), comprovando que o cliente António Silva adquiriu uma unidade de cada um dos artigos anteriormente criados, A1 e A2, pelos preços de 1.500 CVE e 1.800 CVE, respectivamente (Figura 45).

OpenERP
Made by Tiny & Axelor

MAIN MENU **SHORTCUTS** Modules

Facturas Search Form Calendar Gantt Graph Process

Save Save & Edit Cancel << First < Previous [11/11] Next > Last >>

Diário: x Sales Journal Tipo: Factura de Clientes ? Numero da factura: 2011/010
 Parceiro: António Silva Endereço de facturação: António Silva, Rua Cid Moeda: CVE
 ? Data de facturação: 2011/01/23 ? Forçar período: Jan. 2011 (manter vazio para usar o período actual)

Factura Outras Informações Pagamentos

? Conta: 110200 Debtors Descrição:
 ? Termo do Pagamento:

Linha de Factura

Descrição	Conta	Quantidade	Unidade de medida	Preço Unitário	Desconto (%)	Subtotal
[A1] Rato Wireless	100000 Fixed Asset Account	1,00	PCE	1 500,00	0,00	1 500,00
[A2] Mala p/ Portátil	100000 Fixed Asset Account	1,00	PCE	1 800,00	0,00	1 800,00

Import | Export << First < Previous [1 - 2 of 2] Next > Last >>

Taxes

Descrição do imposto	Base	Montante
x VAT	3 300,00	495,00

Sem imposto: 3 300,00 Tax: 495,00
 ? Pago/Conciliado: ☐ Total: 3 795,00
 Estado: Abrir ? Residual: 3 795,00

Figura 45: Representação da Factura emitida

Uma vez que foi emitida uma factura a crédito, suponhamos que o cliente regressou dias depois para liquidar a dívida, utilizando para tal o cheque como modo de pagamento.

A implementação desta funcionalidade exigiu a criação de um mapeamento com os códigos dos modos de pagamento criados no ERP Primavera (Figura 46):

Modos de Pagamento

Descrição: Código Primavera:

Diário:

Figura 46: Criação e mapeamento dos Modos de Pagamento

No ERP Primavera, o modo de pagamento com o código DEP, corresponde a Recebimentos por Cheque.

A acção do cliente dará origem à emissão de um documento de Recebimento de Cliente, conforme comprova a Figura 47:

Recibos

Cliente: Data:

Numero de cheque/Trans.: Metodo de Pagamento:

Linhas de Recibo

Factura	Total	Pago	Residual
CI: 2011/010	3 795,00	3 795,00	0,00

Import | Export

State:

Figura 47: Emissão do Recibo do cliente C1

Conforme é possível constatar, o recibo liquidou a factura com o número 2011/010, que foi anteriormente emitida para o cliente com o código C1.

No OpenERP, o último passo corresponde à exportação de todos os movimentos gerados num determinado período, o que inclui as fichas dos artigos e dos clientes e as facturas e recibos de clientes emitidos.

A título de exemplo, a Figura 48 ilustra o formulário criado que permite a geração e exportação do ficheiro XML contendo as vendas emitidas num determinado período:

Vendas para o Primavera

Data Inicial: Data Final:

Exportacao Terminada

839.00 bytes

Figura 48: Formulário de exportação das vendas emitidas

De acordo com a especificação inicial, os nomes dos ficheiros são criados automaticamente, em conformidade com a nomenclatura pré definida, que obedece a determinados critérios, permitindo facilmente a identificação do tipo de ficheiro e a sua data de emissão (Figura 49):

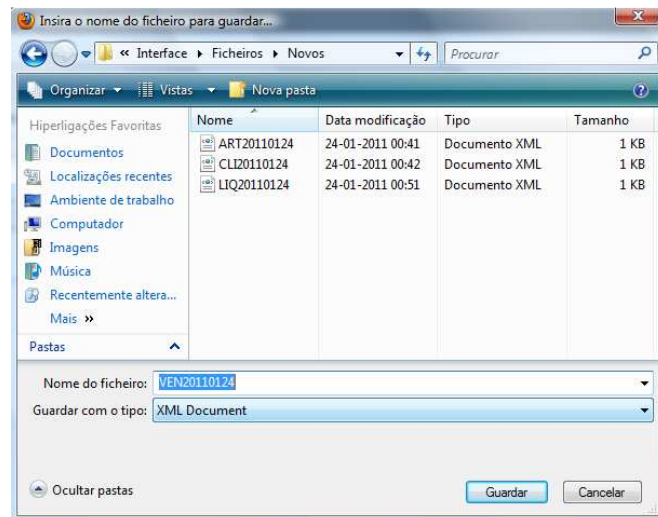


Figura 49: Exportação do ficheiro de vendas

O conteúdo do ficheiro VEN20110124.xml será o seguinte:



Figura 50: Conteúdo do ficheiro de vendas exportado pelo OpenERP

4.7.2 Operações a serem realizadas no ERP Primavera

Após a exportação, a partir do OpenERP, dos ficheiros pretendidos, será necessário importá-los para o ERP Primavera. Para tal, será necessário aceder à interface desenvolvida, cujo *layout* está representado na Figura 51:

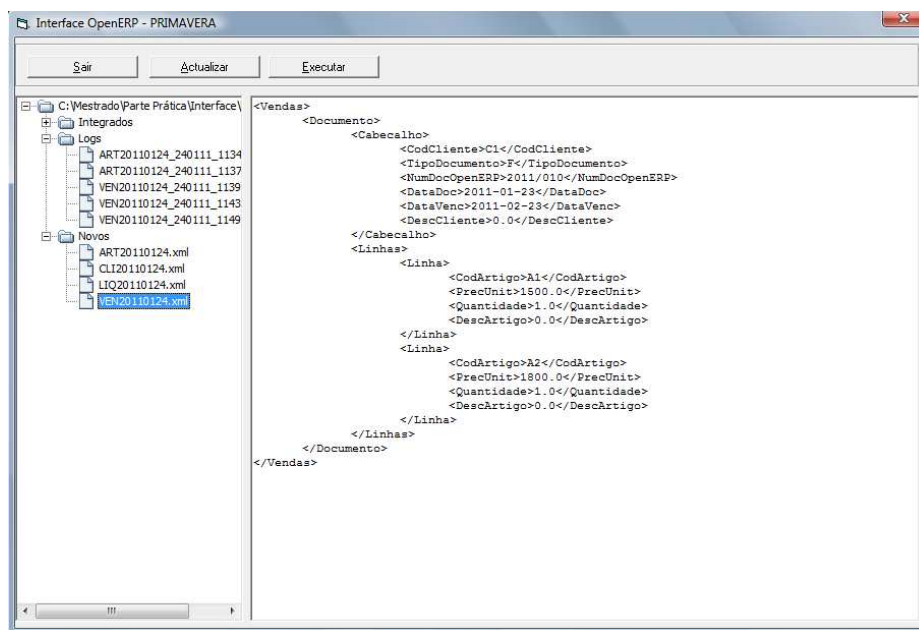


Figura 51: Layout do integrador desenvolvido

Conforme é possível verificar, o integrador consegue ler e também pré-visualizar os ficheiros que foram exportados para a pasta "...\\Novos". Além disso, permite guardar um histórico de todos os ficheiros que foram integrados com sucesso ("...\\Integrados") ou aqueles que, por algum motivo, não foram integrados ("...\\Logs"). Neste último caso, é criado um ficheiro de texto com o *Log*, cujo conteúdo também pode ser visualizado através do integrador (Figura 52).

Ao carregar no botão Executar, o utilizador é confrontado com a confirmação da operação e, em caso de resposta afirmativa, os ficheiros presentes na pasta "...\\Novos" serão processados. Em caso de sucesso, serão movidos para a pasta "...\\Integrados". Caso contrário, permanecerão na pasta "...\\Novos" e, consequentemente, será criado o respectivo ficheiro de *Log*.

O erro apresentado na Figura 52, deriva do facto de existirem caracteres inválidos no conteúdo do ficheiro XML especificado. Neste caso, trata-se da utilização de um carácter com uma acentuação, na descrição do artigo A2, "Mala p/ portátil".

Uma possível solução para o problema passa pela inclusão de um detalhe na definição do cabeçalho do arquivo XML, onde deve ser especificado o conjunto de caracteres a ser utilizado:

```
<?xml version="1.0" encoding="utf-8"?>
```

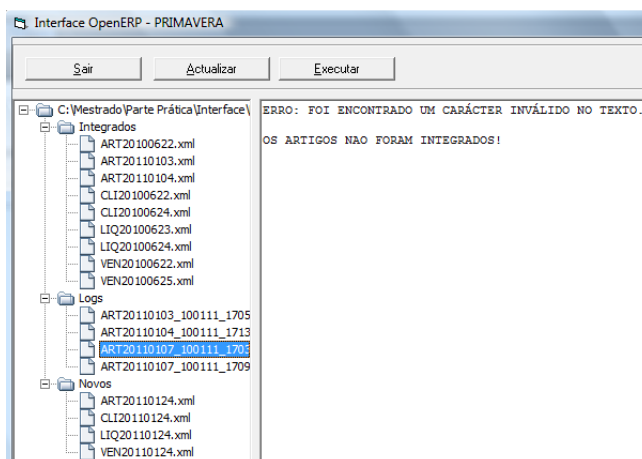


Figura 52: Log – Erro na integração de um ficheiro de Artigos

Porém, a solução acima apontada, não resolve os problemas com os caracteres "&", "<" e ">" e outros caracteres especiais. Por exemplo, o carácter "<" dentro de um elemento XML, irá gerar um erro porque o *parser* o interpreta como o início de um novo elemento.

Uma alternativa para evitar problemas com este tipo de caracteres passa pela utilização do CDATA (*Character Data*), em que, tudo o que estiver dentro de uma secção CDATA será ignorado pelo *parser*.

De seguida, passa-se a apontar outro exemplo de um erro ocorrido na integração do ficheiro de clientes (Figura 53):

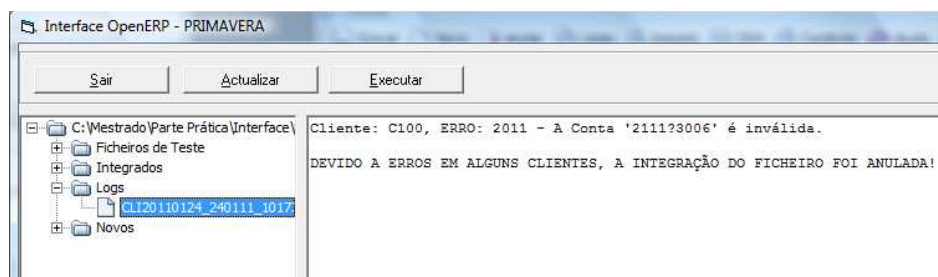


Fig. 53: Log - Erro na integração de um ficheiro de clientes

Este erro está relacionado com as parametrizações que devem ser efectuadas no ERP Primavera para que, no momento da criação das fichas de clientes no módulo de Vendas, estes possam ser automaticamente criados no Plano de Contas da Contabilidade.

No exemplo apresentado, a aplicação está a tentar criar o registo no Plano de Contas, com o código 2111?3006. O problema está no carácter '?' que representa um *wild card* e deve ser substituído por um valor numérico. De acordo com as parametrizações do Plano de Contas da base de dados da empresa utilizada, as contas de clientes devem ser constituídas pelo prefixo 2111 + 4 dígitos, sendo que os últimos 4 dígitos são valores sequenciais, incrementados automaticamente pela aplicação. Entretanto, no exemplo apresentado, o código está a ser constituído por 2111 + 5 dígitos, o que não está coerente com a estrutura do Plano de Contas.

Conforme ilustrado na Figura 54, o campo “Conta Contabilidade” aparece automaticamente preenchido com o valor 3006, no formulário de criação de novas fichas de clientes.

Isto indica que o valor 3006 corresponde aos últimos 4 dígitos da conta do próximo cliente a ser criado no sistema.

The screenshot shows a window titled 'Cliente' with a menu bar containing 'Gravar', 'Novo', 'Anular', 'Listas', 'Imprimir', 'CRM', 'Contexto', 'Ajuda', and 'Cancelar'. Below the menu is a 'Cliente:' label and a search icon. A tabbed interface shows 'Moradas', 'Bancos', 'Dados Fiscais', 'Dados Comerciais', and 'Outros Dados'. The 'Dados Fiscais' tab is active, showing 'Encargos com Títulos', 'Encargos Bancários', and 'Campos do Utilizador'. Below this, there are sections for 'Crédito', 'Observações', 'Retenções', 'Anexos', and 'Contabilidade'. The 'Contabilidade - 2010' section contains three fields: 'Conta Contabilidade' (pre-filled with '3006'), 'Centro de Custo' (empty), and 'Conta Analítica' (empty). At the bottom, there is a 'Segmento Terceiro' dropdown set to '001' and a 'Geral' label.

Figura 54: Ficha de clientes - Preenchimento automático do campo “Conta Contabilidade”

As parametrizações das ligações entre o módulo de Vendas e o módulo de Contabilidade devem ser feitas no Administrador do ERP.

The screenshot shows a window titled 'Gestão Contabilidade - Parâmetros do Exercício' with a dropdown for 'Ano do Exercício' set to '2011'. The 'Abertura Contas' tab is active. On the left, a 'Pastas' tree shows 'Parâmetros' expanded, with sub-items: 'Gerais', 'Bloqueios', 'Contas/Módulos', 'Prefixos de Contas', 'Recapitulativos', 'Abertura Contas' (selected), 'Moeda Alternativa', and 'Pendentes'. The main area is titled 'Indique o perfil das contas a criar automaticamente na abertura de fichas de terceiros'. It contains a table with four columns: 'Clientes', 'Fornecedores', 'Outros devedores', and 'Outros credores'. The first row under 'Clientes' is pre-filled with '2111?777'. At the bottom, there are buttons for 'Confirmar', 'Cancelar', and 'Ajuda'.

Figura 55: Parametrização da criação de contas de Clientes na Contabilidade

Para corrigir a situação apontada, a parametrização da criação automática das contas de clientes deve ser corrigida para o respectivo prefixo (2111), seguido de 4 dígitos (Figura 55).

Após a correcção das anomalias identificadas, já será possível integrar os ficheiros no ERP Primavera. A Figura 56 ilustra a Factura criada no referido ERP. Antes da criação deste documento de venda, o sistema teve de criar as fichas dos artigos A1 e A2 e do cliente C1.

Editor de Vendas

Documento: Factura 2011 1 23-01-2011

Entidade: C1 Antonio Silva Rua Cidade de Lisboa Praia 417

Contribuinte: 945004543 V/Refer. 2011/010

Resumo:

Merc./Serv.:	3.300,00
Descontos:	0,00
IVA:	495,00
Outros:	0,00
Sub-Total:	3.795,00
Acerto:	0,00
Ecovalor:	0,00
Total: CVE	3.795,00
Stock:	0,00

Artigo	Arm.	Lote	Descrição	IVA	Pr. Unit.	Desc.	UN	Quant.	Total Liq.	F.Conv.	IVA Ded.	IVA não c.
A1			Rato Wireless	15,00	1.500,00	0,00	UN	1,00	1.500,00	1,00	100,00	0
A2			Mala p. portatil	15,00	1.800,00	0,00	UN	1,00	1.800,00	1,00	100,00	0

Figura 56: Detalhes da Factura integrada no ERP Primavera

Adicionalmente, na Figura 57, pode-se confirmar a criação dos registos na base de dados Primavera, através de *queries* criadas com o Microsoft SQL Server.

SQLQuery1.sql - E...RIDEMO (sa (61))*

```

select Artigo, Descricao, UnidadeVenda from Artigo where Artigo = 'A1' or Artigo = 'A2'
select Cliente, Nome, Fac_Mor, Fac_Local, Fac_Cp from Clientes where Cliente = 'C1'
select TipoDoc, NumDoc, Entidade, Data, TotalMerc, TotalIva from CabecDoc where TipoDoc = 'FA' and serie = '2011' and NumDoc = '1'
select TipoEntidade, Entidade, TipoDoc, DataDoc, ValorRec from CabLiq where TipoEntidade = 'C' and Entidade = 'C1'
  
```

Results

Artigo	Descricao	UnidadeVenda
1 A1	Rato Wireless	UN
2 A2	Mala p. portatil	UN

Cliente	Nome	Fac_Mor	Fac_Local	Fac_Cp
1 C1	Antonio Silva	Rua Cidade de Lisboa	Praia	417

TipoDoc	NumDoc	Entidade	Data	TotalMerc	TotalIva
1 FA	1	C1	2011-01-23 00:00:00.000	3300	495

TipoEntidade	Entidade	TipoDoc	DataDoc	ValorRec
1 C	C1	RE	2011-01-25 00:00:00.000	3795

Figura 57: Apresentação dos dados integrados no ERP Primavera

Conforme foi referido anteriormente, a exequibilidade da integração dos documentos com o módulo de Contabilidade, é garantida pelas funcionalidades do próprio ERP. Neste contexto, a Figura 58 mostra o lançamento contabilístico resultante da integração da factura na Contabilidade.

Movimentos para a Contabilidade e Bancos

Documento: Factura

Número: 1

Diário: 51

Vendas a Credito

Opções

Recalcular Mov.

Ordem/ Geral/ Analítica

Centros de Custo

Funções

Fluxos Caixa

Moeda Lançamento: CVE

Visualização em: CVE EUR

Taxa de Câmbio: CVE/CVE

1,0000000

CVE/EUR

110,2650000

Conta	Débito (CVE)	Crédito (CVE)	IVA	% n/ Dec	IVA Autolic
21113006	3.795				
2442121		495			
7111		3.300			

Terceiro:

Tipo Op.:

Geral	3.795,00D	3.795,00C	[1] Antonio Silva
Analítica	0,00D	0,00C	
C. Custo	0,00D	0,00C	

Recalcul.

0,00

0,00

Confirmar

Cancelar

Ajuda

Figura 58: Integração da Factura na Contabilidade

5 Conclusões e trabalho futuro

Este capítulo está dividido em duas secções, consistindo a primeira numa conclusão genérica, e a segunda, numa abordagem a um conjunto de evoluções ou melhorias que podem ser implementadas, com o intuito de enriquecer e expandir o âmbito do trabalho realizado.

5.1 Considerações

Actualmente, a heterogeneidade de SI existente nas organizações é um problema recorrente, que normas como o XML vieram simplificar, oferecendo características que facilitam a integração entre os diferentes sistemas.

Em Cabo Verde, o ERP da Primavera BSS e, mais recentemente, o OpenERP, têm vindo a ganhar grande preponderância no mercado e, como tal, têm sido adoptados por diversas organizações, pertencentes às mais variadas áreas de negócio.

Neste contexto, o grande objectivo deste trabalho consistiu na apresentação de uma solução baseada no desenvolvimento de uma arquitectura, suportada por trocas de ficheiros XML, que possibilitasse a integração entre esses dois sistemas.

Para tal, foi necessário fazer um enquadramento da situação actual e das tendências da integração de sistemas, sustentado por uma revisão da literatura relacionada com os SI, sistemas ERP e Integração de SI. Ainda no âmbito dos aspectos teóricos abordados, foi apresentado um caso de estudo que envolve a integração de dois sistemas mundialmente conhecidos e líderes de mercado nas respectivas áreas, quais sejam o ERP da SAP e o Sistema de Gestão de Viagens da Amadeus.

A escolha do *standard* XML como base para a integração, deveu-se à sua abrangência e flexibilidade, no que diz respeito ao manuseamento e integração de informação.

Em suma, recupera-se nesta secção os objectivos inicialmente delineados, para se referir que foram atingidos na sua generalidade:

- ✓ A integração da informação passa a ser feita de forma automática e, não havendo intervenção humana nesse processo, a probabilidade de ocorrência de erros é menor;
- ✓ Redução dos custos operacionais;
- ✓ Obtenção de uma maior rentabilidade do negócio.

No entanto, na próxima secção é proposto um conjunto de melhorias que poderão ser implementadas, com vista à optimização e maior abrangência do trabalho desenvolvido.

5.2 Trabalho Futuro

De forma a tirar melhor proveito da arquitectura desenvolvida, propõe-se as seguintes melhorias ou funcionalidades:

- ✓ Desenvolver a arquitectura recorrendo à tecnologia de *Web Services*;
- ✓ Alargar o âmbito da integração para outros módulos dos ERP, nomeadamente, Gestão de Stocks, Tesouraria e Compras;
- ✓ Desenvolver mecanismos que permitam que a integração seja bidireccional, o que obrigaria a uma reengenharia da solução;
- ✓ No OpenERP, criar uma *flag* que permita controlar/distinguir os documentos exportados e os não exportados;
- ✓ Permitir que os ficheiros sejam gerados a partir de um processo automático no sistema OpenERP, que os depositará numa directoria específica (a versão desenvolvida apenas contempla a exportação manual, ou seja, é o utilizador quem dá ordem ao sistema para exportar os ficheiros);
- ✓ Para evitar que os dados possam ser facilmente acedidos através da pasta “Integrados”, o que representa uma falha de segurança, pode-se implementar uma das seguintes alternativas:
 - i. Eliminar a pasta “Integrados” na estrutura de pastas, passando a existir apenas as pastas “Novos” e “Logs”;
 - ii. Recorrer a técnicas adequadas (ex: Framework da Microsoft⁹) para encriptar os ficheiros XML contidos na pasta “Integrados”.

⁹ <<http://msdn.microsoft.com/en-us/library/ms229749.aspx>> Acesso em 23 de Março de 2011.

Referências bibliográficas

- Amaral, J. F. F., *Conceptualização de uma Aplicação CRM para o Serviço Pós-Venda*, Aveiro: Universidade de Aveiro, 2009.
- Azevedo, C. e Serdeira, P., *Os ERP's (Enterprise Resource Planning) como soluções integradas para a indústria da Hotelaria e Turismo, «Sistemas de Informação»*, 2001.
- Cameira, R., *Sistemas Integrados de Gestão – Perspectivas de evolução e questões associadas*, Rio de Janeiro, 2001.
- Carvalho, J. A., *MISW – Uma Metodologia de Integração com Serviços Web*, Universidade de Fortaleza - UNIFOR, 2003.
- Cassarro, A., *Sistemas de informações para tomada de decisões*, Thomson Pioneira, 2004.
- Christensen, E. *et al.*, *Web Services Description Language 1.1.*, 2004. Disponível em: <<http://www.w3.org/TR/wsdl>>. (Acedido em 11/12/2010)
- Custódio, D. L. A., *Gestão da Heterogeneidade Sob o Ponto de Vista Tecnológico*, Universidade de Coimbra, 2006.
- Dantas, M., *Computação Distribuída*, UFSC, Departamento de Informática e Estatística. Disponível em: <http://www.inf.ufsc.br/~mario/notasdeaulacd.pdf>>. (Acedido em 10-10-2010)
- Davenport, T. H e Prusak, L., *Ecologia da informação*, São Paulo, Futura, 2000.
- De Sordi, J.O. e Marinho, B.L., *Integração entre Sistemas: Análise das Abordagens Praticadas pelas Corporações Brasileiras*, RBGN, São Paulo, 2007.
- Decreto-Lei n.º 5/2008, de 4 de Fevereiro, publicado no Boletim Oficial, I Série, da República de Cabo Verde.
- Esteves, J., e Jorge, J., *Análise Comparativa de Metodologias de Implantação de SAP*, Évora, 2001.
- Favaretto, F., *Integração de sistemas utilizando Data Warehouse: possibilidades para sistemas de informação de controle de estoques*, XII SIMPEP, 2006.

- Fontana, R. M., Lima, L. C. M., Machado, F. P., Krainski, A. F. C. e Misugi, J., Análise da Adequação de um ERP Livre a uma Empresa Brasileira, XXVIII Encontro Nacional de Engenharia de Produção, Rio de Janeiro, 2008.
- Garcia, S. S. e Shinotsuka, T. H., EAI – Enterprise Application Integration, Universidade Federal do Rio de Janeiro, 2001.
- Gonçalves, C. e Fernandes, J. S., Sistema de Normalização Contabilística proposto pela CNC enquanto referencial normativo em Cabo Verde, 2009.
- Gonçalves, J. E. L., As empresas são grandes coleções de processos, RAE Revista de Administração de Empresas, São Paulo, 2000.
- Haberkorn, E., Teoria do ERP (Enterprise Resource Planning), São Paulo, Makron Books, 1999.
- Hernandez, J., The SAP R/3 Handbook, McGraw-Hill, 2000.
- Laudon, K. C. e Laudon, J. P., Management Information Systems – Managing The Digital Firm, 9ª Edição, Pearson – Prentice Hall, New Jersey, 2006.
- Licht, F. L., Pereira, A., Reis, E. e Schulze, B., Objetos Distribuídos e Java, Laboratório Nacional Computação Científica LNCC, Petrópolis - RJ.
- Linthicum, D. S., Enterprise Application Integration, Addison Wesley, Massachusetts, 2000.
- Linthicum, D. S., Enterprise Application Integration, Saga Software, 1998.
- Mainetti Jr., S., Objetos Distribuídos, Visionnaire, Curitiba, 1997.
- Manga, M., Ferreira P., Sousa, P. e Loureiro, V., SAP R/3 & SAP Netweaver, FEUP, 2006.
<http://twiki.fe.up.pt/twiki/pub/ERSS0607/G7_DocumentosdeApoioaoDocumentoRequisitos/SAPR3SAPNetweaver.pdf>, (Acedido em 10/09/2010).
- Marques, J. A., A integração de SI é uma questão de Alinhamento de Processos e Sistemas, Entrevista - Cadernos Link, 2004.
- MARQUES, J. A., Service Oriented Architecture (SOA): uma tecnologia ou arquitectura, Lisboa, 2005.
- Martins, V., Integração de Sistemas de Informação – Perspectivas, Normas e Abordagens, Edições Silabo, 2006.

- Matos, T., Introdução ao COM – Component Object Model, ISEP: Instituto Superior de Engenharia do Porto, 2004.
- Mcgee, J. V e Prusak, L., Gerenciamento estratégico da informação, Rio de Janeiro, Campus, 1994.
- Silva, M. M., Integração de Sistemas de Informação, FCA, 2003.
- Mülbert, A. L. e Ayres, N. M., Fundamentos para Sistemas de Informação, 2ª Edição, Universidade do Sul de Santa Catarina, 2005.
- Silva, F. O., Integração de Sistemas e Plataformas como Solução para a Gestão da Informação de Clientes, Porto, ISEP, 2004.
- Pinckaers, F. e Gardiner, G., OpenERP - A Modern Approach to Integrated Business Management, Release 1.0, 2009.
- Pires, T. M., Sistemas de Informação na Indústria – O caso do Izaro Grey, Aveiro: Universidade de Aveiro, 2008.
- Primavera Academy, Introdução ao VBA para Primavera, 2008.
- Primavera Business Software Solutions, Manual de Extensibilidade do ERP Primavera, 2007.
- Primavera Business Software Solutions, Manual de Extensibilidade II, 2009.
- Ribeiro, V., Implementação de um ERP numa PME: o “antes” e o “depois”, Aveiro: Universidade de Aveiro, 2009.
- Ruh, W. A., Maginnis F. X. & Brown, W. J., Enterprise Application Integration: a Wiley Tech Brief, Wiley, New York, 2001.
- Santos, A. A., Kaldeich, A., e Silva, L., Sistemas ERP: Um enfoque sobre a utilização do SAP R/3 em contabilidade e custos, ENEGEP, 2003.
- SAP Netweaver – Solution Overview.
<http://www.sap.com/platform/netweaver/pdf/BWP_OV_SAP_NetWeaver.pdf>
(Acedido em 14/09/2010).
- Seruca, I. e Teixeira, L., Sistemas de Informação em Gestão – Fundamentos organizacionais dos Sistemas de Informação, Aveiro: Universidade de Aveiro, 2006.

- Stair, R. e Reynolds, G., Principles of Information Systems, 8ª Edição, Thomson - Course Technology, Canadá, 2008.
- Stair, R. M., Princípios de sistemas de informação: uma abordagem gerencial, Rio de Janeiro, LTC, 2002.
- Teixeira, L., Contribuições para o Desenvolvimento de Sistemas de Informação na Saúde: Aplicação na área da Hemofilia, Aveiro: Universidade de Aveiro, 2009.
- Varajão, J., Arquitectura da Gestão de Sistemas de Informação, 3ª Edição Actualizada, FCA, 2005.
- William, G.C., Implementing SAP ERP Sales & Distribution - Essencial Skills for SAP Professionals, McGraw-Hil, 2008.
- Zaidan, F. H., Sistemas de Informações Empresariais: Integração de Sistemas Interorganizacionais utilizando troca de documentos eletrônicos via XML, Belo Horizonte, 2006.